Database Administrator's Guide (Printable)

Added by K-2, last edited by K-2 on Feb 12, 2010

Sun Role Manager 5.0.3 Database Administrator's Guide

About This Guide

This guide describes the Sun™ Role Manager 5.0.3 database schema.

Who Should Read This Guide

The *Sun Role Manager 5.0.3 Database Administrator's Guide* is written for database administrators who are responsible for supporting the Sun Role Manager software at the database level.

- For information about installing and readying your database server to support Sun Role Manager, see the Preparing to Install Sun Role Manager chapter in the *Sun Role Manager 5.0.3 Installation and Upgrade Guide*.
- For information about downloading JDBC™ drivers, as well as how to create the Role Manager schema on the database server, see the Installing Sun Role Manager chapter in the *Sun Role Manager 5.0.3 Installation and Upgrade Guide*.

GlobalUsers Module

Tables in the GlobalUsers Module

This chapter describes the tables that make up the Sun™ Role Manager software (Role Manager) GlobalUsers module.

GLOBALUSERS Table

Structure

	GLOBALUSERS
PK	**GLOBALUSERKEY**
I1	USERNAME FIRSTNAME LASTNAME MIDDLENAME STREET CITY STATEORPROVINCE ZIPORPOSTALCODE COUNTRYORREGION FAX PHONE EXTENSION MOBILE PAGER TITLE PRIMARYEMAIL SECONDARYEMAIL OFFICENAME DESCRIPTION COMMENTS STATUSKEY SUSPENDEDDATE ENABLEDDATE DISABLEDDATE DELETEDATE USERDATA EMPLOYEEID CUSTOMPROPERTY1 CUSTOMPROPERTY2 CUSTOMPROPERTY3 CUSTOMPROPERTY4 CUSTOMPROPERTY5 CUSTOMPROPERTY6 CUSTOMPROPERTY7 CUSTOMPROPERTY8 CUSTOMPROPERTY9 CUSTOMPROPERTY10 CUSTOMPROPERTY11 CUSTOMPROPERTY12 CUSTOMPROPERTY13 CUSTOMPROPERTY14 CUSTOMPROPERTY15 CUSTOMPROPERTY16 CUSTOMPROPERTY17 CUSTOMPROPERTY18 CUSTOMPROPERTY19 CUSTOMPROPERTY20 CREATEUSER UPDATEUSER CREATEDATE UPDATEDATE SRMCREATEDATE SRMUPDATEDATE EMPLOYEETYPE SERVICEDESKTICKETNUMBER STARTDATE ENDDATE MANAGER BUSINESSAPPROVER TECHNICALAPPROVER DELEGATE LOCATION JOBCODES

Indexes

Index	PK	Unique	Keys

PK_GLOBALUSERS	✓	Yes	GLOBALUSERKEY
IX_GU_UNAME		No	USERNAME

Description

Users that are stored in the Role Manager Identity Warehouse are saved in the GLOBALUSERS table. Each user is known as a global user and identified by a unique GLOBALUSERKEY. This table stores basic information about every user in the USERNAME, FIRSTNAME, LASTNAME, and MIDDLENAME fields. There are also fields to save additional information about the user, including the user's address, email, phone, and various other custom fields. Because the user can be associated with a workflow, fields are provided to save the user's manager, business approver, and technical approver. A DELEGATE field is also present so that the user can specify a delegate. The STATUSKEY field contains the user's status. For audit purposes the table carries the account's creation date and update date, the user who created the account, and the name of the user who updated the account.

Primary Keys

1.PK_GLOBALUSERS - primary key on column GLOBALUSERKEY

Foreign Keys

None

Indexes

1.IX_GU_UNAME - non-unique index on column USERNAME

top

GLOBALUSER_ACCOUNTS TABLE

Structure

	GLOBALUSER_ACCOUNTS
PK,I1 **PK,U1**	**GLOBALUSERKEY** **ACCOUNTKEY**

Indexes

Index	**PK**	**Unique**	**Keys**
PK_USER_ACCOUNTS	✓	Yes	GLOBALUSERKEY, ACCOUNTKEY
IX_GLOBALUSER_ACCOUNTS_ACCOUNT		Yes	ACCOUNTKEY
IX_GU_ACCTS_GUID		No	GLOBALUSERKEY

Description

A Global user in Role Manager is associated with an account by associating the GLOBALUSERS and ACCOUNTS tables with a derived table, GLOBALUSER_ACCOUNTS. This table carries the ACCOUNTKEY and GLOBALUSERKEY.

Primary Keys

1.PK_USER_ACCOUNTS - composite primary key on column GLOBALUSERKEY and ACCOUNTKEY

Foreign Keys

None

Indexes

1.IX_GU_ACCTS_GUID - non-unique index on column GLOBALUSERKEY
2.IX_GLOBALUSER_ACCOUNTS_ACCOUNT - unique constraint on column ACCOUNTKEY

top

GLOBALUSER_ROLES Table

Structure

	GLOBALUSER_ROLES
PK	**ID**
I1	GLOBALUSERKEY
I2	ROLEKEY
I4	SERVICEDESKTICKETNUMBER
I3	STARTASSOCIATIONDATE
	ENDASSOCIATIONDATE
	START_REQUEST_ID
	END_REQUEST_ID
	STATUS_ID
	ROLE_VERSION_ID
	MODIFIED_BY
	MODIFIED_DATE
	MODIFICATION_GROUP_ID
	REQUESTED_MODIFICATION

Indexes

Index	**PK**	**Unique**	**Keys**
PK_GLOBALUSER_ROLES	✓	Yes	ID
IX_GU_ROLES_GUID		No	GLOBALUSERKEY
IX_GU_ROLES_RID		No	ROLEKEY
IX_GU_ROLES_RQSTART		No	START_REQUEST_ID
IX_GU_ROLES_RQEND		No	END_REQUEST_ID

Description

Each Global user can be associated with a role and this association is represented in a derived table, GLOBALUSER_ROLES. This table references tables GLOBALUSERS and ROLES using the GLOBALUSERKEY and ROLEKEY fields.

Primary Keys

1.PK_GLOBALUSER_ROLES - composite primary key on columns GLOBALUSERKEY and ROLEKEY

Foreign Keys

None

Indexes

1.IX_GU_ROLES_GUID - non-unique index on column GLOBALUSERKEY
2.IX_GU_ROLES_RID - non-unique index on column ROLEKEY
3.IX_GU_ROLES_RQSTART - non-unique index on column START_REQUEST_ID
4.IX_GU_ROLES_RQEND - non-unique index on column END_REQUEST_ID

top

GLOBALUSERSTATUSES Table

Structure

	GLOBALUSERSTATUSES
PK	**STATUSKEY**
	LABEL SORTORDER

Indexes

Index	PK	Unique	Keys
PK_GLOBALUSERSTATUSES	✓	Yes	STATUSKEY

Description

Similar to the Business Unit, each user has a status associated with it. The GLOBALUSERSTATUSES table saves the possible statuses that can be associated with the globalusers. The actual 'status' is represented in the STATUSKEY and LABEL fields.

Primary Keys

1.PK_GLOBALUSERSTATUSES - primary key on column STATUSKEY.

Foreign Keys

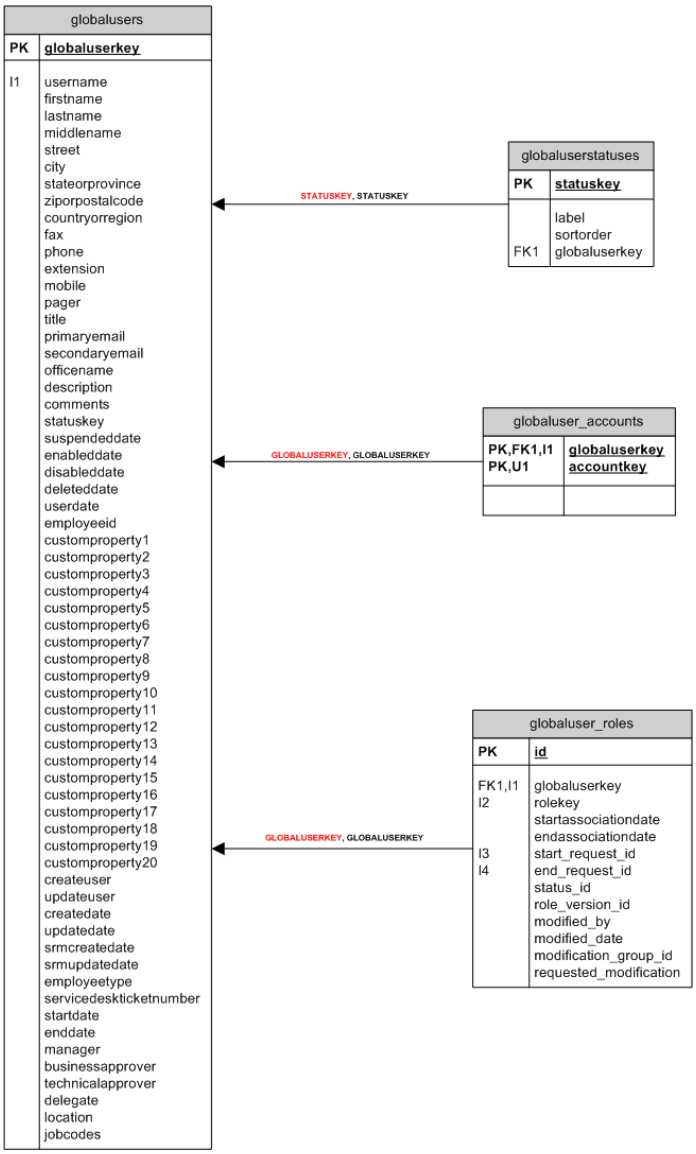
None

Indexes

None

top

GlobalUsers Module Entity Relationship



top

Business Structure Module

Tables in the Business Structure Module

This chapter describes the tables that make up the Role Manager Business Structure module.

BUSINESSUNITS Table

Structure

	BUSINESSUNITS
PK	BUSINESSUNITKEY
	BUSINESSUNITNAME
	MAINPHONE
	OTHERPHONE
	FAX
	EMAIL
	WEBSITE
	STREET1
	STREET2
	STREET3
	CITY
	STATEORPROVINCE
	ZIPORPOSTALCODE
	COUNTRYORREGION
	DIVISION
	CREATEUSER
	UPDATEUSER
	CREATEDATE
	UPDATEDATE
	BUSINESSUNITTYPE
	BUSINESSUNITOWNER
	BUSINESSUNITADMINISTRATOR
	BUSINESSUNITCODE
	BUSINESSUNITDESCRIPTION
	MAILCODE
	SERVICEDESKTICKETNUMBER

STATUSKEY
BUSINESSUNITMANAGERS

Indexes

Index	PK	Unique	Keys
PK_BUSINESSUNITS	✓	Yes	BUSINESSUNITKEY

Description

The various departments in an organization are listed as Business Units in Role Manager. Each Business Unit has users associated with it, similar to users listed in a department. The BUSINESSUNITS table in Role Manager carries the information for the various departments in an organization. A unique BUSINESSUNITKEY identifies each business unit.

The various fields listed under the table provide more information on the business units such as the BUSINESSUNITNAME, PHONE, EMAIL, WEBSITE, CITY, STATE, and ZIP. For audit purposes, the Business Unit's creation time, the user who created the business unit, the update time, and the user who updated the Business unit are noted. Each Business unit is assigned an owner and this information is stored in the BUSINESSUNITOWNER field. In addition, there are various business unit managers who are assigned to each business unit and they are entered in the BUSINESSUNITMANAGERS field. These managers can be stored in a comma-separated format. If a Business Unit has an Administrator, the Administrator can be assigned in the BUSINESSUNITADMINISTRATOR field.

Primary Keys

1.PK_ BUSINESSUNITS - primary key on column BUSINESSUNITKEY

Foreign Keys

None

Indexes

None

top

BU_GLOBALUSERS Table

Structure

	BU_GLOBALUSERS
PK,I1	BUSINESSUNITKEY
PK,I2	GLOBALUSERKEY

Indexes

Index	PK	Unique	Keys
PK_BU_GLOBALUSERS	✓	Yes	BUSINESSUNITKEY, GLOBALUSERKEY
IX_BU_GLOBALUSERS_BUID		No	BUSINESSUNITKEY
IX_BU_GLOBALUSERS_GUID		No	GLOBALUSERKEY

Description

The association between a user and Business Unit is shown in a derived table, BU_GLOBALUSERS. This table carries the BUSINESSUNITKEY and GLOBALUSERKEY.

Primary Keys

1.PK_BU_GLOBALUSERS - composite primary key on columns BUSINESSUNITKEY and GLOBALUSERKEY

Foreign Keys

None

Indexes

1.IX_BU_GLOBALUSERS_BUID - non-unique index on column BUSINESSUNITKEY
2.IX_BU_GLOBALUSERS_GUID - non-unique index on column GLOBALUSERKEY

top

BU_ROLES Table

Structure

	BU_ROLES
PK	BUSINESSUNITKEY
PK	ROLEKEY
PK	ROLE_VERSION_ID

Indexes

Index	PK	Unique	Keys
PK_BU_ROLES	✓	Yes	BUSINESSUNITKEY, ROLEKEY, ROLE_VERSION_ID

Description

The association between a role and the Business Unit is shown in a derived table, BU_ROLES. The fields BUSINESSUNITKEY and ROLEKEY are used to reference the BUSINESSUNITS and ROLES tables respectively.

Primary Keys

1.PK_BU_ROLES - composite primary key on columns BUSINESSUNITKEY and ROLEKEY

Foreign Keys

None

Indexes

None

top

BU_POLICIES Table

Structure

	BU_POLICIES
PK	BUSINESSUNITKEY
PK	POLICYKEY
PK	POLICY_VERSION_ID

Indexes

Index	PK	Unique	Keys
PK_BU_POLICIES	✓	Yes	BUSINESSUNITKEY, POLICYKEY, POLICY_VERSION_ID

Description

The association between a policy and the Business Unit is shown in a derived table, BU_POLICIES. The fields BUSINESSUNITKEY and POLICYKEY are used to reference the BUSINESSUNITS and POLICIES tables respectively.

Primary Keys

1.PK_BU_POLICIES - composite primary key on columns BUSINESSUNITKEY, POLICYKEY, and POLICY_VERSION_ID

Foreign Keys

None

Indexes

None

top

BUHIERARCHY Table

Structure

	BUHIERARCHY
PK	BUSINESSUNITKEY
PK	PARENTBUSINESSUNITKEY

Indexes

Index	PK	Unique	Keys
PK_BUHIERARCHY	✓	Yes	BUSINESSUNITKEY, PARENTBUSINESSUNITKEY

Description

The various Business Units in Role Manager can be represented in a hierarchical format and the hierarchy is represented in the BUHIERACHY table. Each Business Unit has a Parent Business Unit associated with it and this is represented in the PARENTBUSINESSUNITKEY column.

Primary Keys

1.PK_BUHIERARCHY - composite primary key on columns BUSINESSUNITKEY and PARENTBUSINESSUNITKEY

Foreign Keys

None

Indexes

None

top

BUSINESSUNITSTATUSES Table

Structure

	BUSINESSUNITSTATUSES
PK	STATUSKEY
	LABEL

Indexes

Index	PK	Unique	Keys
PK_BUSINESSUNITSTATUSES	✓	Yes	STATUSKEY

Description

A Business Unit in Role Manager can be listed as Active / Inactive and these statuses are listed under the BUSINESSUNITSTATUSES table.

Primary Keys

1.PK_BUSINESSUNITSTATUSES - primary key on column STATUSKEY

Foreign Keys

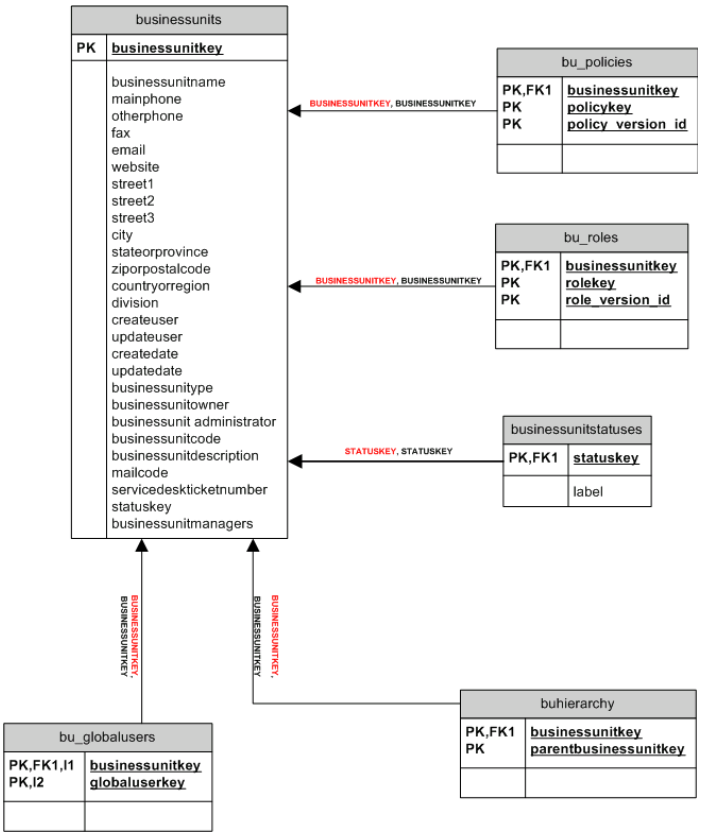
None

Indexes

None

top

Business Structure Module Entity Relationship



top

Resource Types Metadata Module

Tables in the Resource Types Metadata Module

This chapter describes the tables that make up the Role Manager Resource Types Metadata module.

NAMESPACES Table

Structure

	NAMESPACES
PK	NAMESPACEKEY
	NAMESPACENAME NAMESPACESHORTNAME NAMESPACECOMMENTS MULTIVALUE_DELIMETER TYPE_LDAP

Indexes

Index	PK	Unique	Keys
PK_NAMESPACES	✓	Yes	NAMESPACEKEY

Description

The NAMESPACES table lists resource types from the provisioning system that Role Manager connects to, or it lists the resource types that Role Manager connects to otherwise. Each resource type is identified by a unique key called NAMESPACEKEY. In addition, this table also stores its corresponding NAMESPACENAME, NAMESPACESHORTNAME, and NAMESPACECOMMENTS.

Primary Keys

1. PK_NAMESPACES on column NAMESPACEKEY

Foreign Keys

None

Indexes

None

top

ATTRIBUTECATEGORIES Table

Structure

	ATTRIBUTECATEGORIES
PK PK	ATTRIBUTECATEGORYKEY NAMESPACEKEY
	ATTRIBUTECATEGORYNAME ATTRIBUTECATEGORYORDER

Index

Index	PK	Unique	Keys
PK_ATTRIBUTECATEGORIES	✓	Yes	ATTRIBUTECATEGORYKEY, NAMESPACEKEY

Description

An account has various attributes defined that are clustered under a particular category. For example, in Active Directory a user's Exchange Attributes are grouped under the 'Exchange' category, and in Top Secret a user's TSO attributes are grouped under the 'TSO' category. Sun Role Manager stores and handles these different categories under the ATTRIBUTECATEGORIES table. The attribute categories are listed under their particular Resource Types by referencing the NAMESPACES table with the NAMESPACEKEY. The Attribute Category is also defined in a particular order to facilitate the import of accounts.

Primary Keys

1. PK_ATTRIBUTECATEGORIES - composite key on columns ATTRIBUTECATEGORYKEY and NAMESPACEKEY

Foreign Keys

None

Indexes

None

top

ATTRIBUTES Table

Structure

	ATTRIBUTES
PK, I1 PK	**ATTRIBUTEKEY** **ATTRIBUTECATEGORYKEY**
	NAME DESCRIPTION MINVALUE MAXVALUE DEFAULTVALUE MINLENGTH MAXLENGTH EDITTYPE SPACEALLOWEDIN MULTIVALUE ATTRIBUTECASE ATTRIBUTEVALUES EXCLUDEDVALUES LABEL HIDDEN MANDATORY MANAGED ATTRIBUTEORDER ISAUDITABLE ISIMPORTABLE ISMINABLE ISENTITLEMENT_MINABLE ISCERTIFIABLE CLASSIFICATIONS

Index

Index	PK	Unique	Keys
PK_ATTRIBUTES	✓		ATTRIBUTECATEGORYKEY, ATTRIBUTEKEY

Description

The attributes for different Resource Types are listed under the ATTRIBUTES table. Each attribute is mapped to a particular attribute category by way of a reference to the ATTRIBUTECATEGORYKEY from the ATTRIBUTECATEGORIES table. Each attribute is defined by its name and other attribute values such as minimum value, maximum value, excluded value, and default value. The attribute can also be listed as hidden. An edit type lists the data type of the attribute and the label field specifies the attribute name that would be seen in different modules of Role Manager. In addition, a set of flags are assigned in this table to facilitate the handling of the attribute in Role Manager. The functions of these flags are listed below:

- HIDDEN-> The attribute is hidden on the Role Manager UI pages
- MANAGED-> This flag is selected when any operation is to be done on the attribute
- MANDATORY-> This flag specifies that all the operations are to be done on the attribute
- ISIMPORTABLE -> The attribute can be imported from a provisioning system
- ISAUDITABLE -> This flag specifies that auditing can be done on the attribute
- ISMINABLE -> This flag is selected when the attribute is defined for RoleEngineering
- ISENTITLEMENT_MINABLE -> This flag specifies that the attribute's entitlement are defined for Role Engineering
- ISCERTIFIABLE -> Select this flag when certification is to be carried out on the attribute

Primary Keys

1. PK_ATTRIBUTECATEGORIES - composite key on columns ATTRIBUTEKEY and ATTRIBUTECATEGORYKEY

Foreign Keys

None

Indexes

None

top

ATTRIBUTE_VALUES Table

Structure

	ATTRIBUTE_VALUES
PK	**ID**
I1	ATTRIBUTE_ID ATTRIBUTE_VALUE

Index

Index	PK	Unique	Keys
PK_ATTRIBUTE_VALUES	✓	Yes	ID

IX_ATTR_VALUES_AID		No	ATTRIBUTE_ID

Description

The ATTRIBUTE_VALUES table stores the actual values of all entitlements/attributes in Role Manager if they are present in the accounts. The ATTRIBUTE_VALUE field stores the value, and the ATTRIBUTE_ID field ties the table to the ID in the ATTRIBUTES table.

Primary Keys

1.PK_ATTRIBUTE_VALUES - primary key on column ID

Foreign Keys

None

Indexes

1. IX_ATTR_VALUES_AID - non-unique index on column ATTRIBUTE_ID

top

ATTRIBUTE_VALUE_METADATA Table

Structure

	ATTRIBUTE_VALUE_METADATA
PK	**ATTRIBUTE_VALUE_ID**
PK	**ENDPOINT_ID**
	HIGH_PRIVILEGED DATA_OWNER_ID DATA_OWNER_NAME CLASSIFICATION DEFINITION COMMENTS

Indexes

Index	PK	Unique	Keys
PK_ ATTRIBUTE_VALUES_MET	✓	Yes	ATTRIBUTE_VALUE_ID, ENDPOINT_ID

Description

Attributes for various resources have values that are not always comprehensible to managers or end users. System Administrators can add comprehensible names for these attributes in Role Manager so that they are more easily understood. The table that defines the mapping between the attribute value and the attribute's comprehensible name is ATTRIBUTE_VALUE_METADATA table. Every entry in this table references the attribute_values and endpoints table by their respective IDs.

Primary Keys

1. PK_ ATTRIBUTE_VALUES_MET - Composite primary key on columns ATTRIBUTE_VALUE_ID and ENDPOINT_ID

Foreign Keys

None

Indexes

None

top

ENDPOINTS Table

Structure

	ENDPOINTS
PK	**ID**
	NAMESPACE_ID ENDPOINT_NAME HOST_NAME HOST_IP DESCRIPTION COMMENTS CREATEUSER UPDATEUSER CREATEDATE UPDATEDATE SERVICEDESKTICKETNUMBER

Index

Index	PK	Unique	Keys
PK_ENDPOINTS	✓	Yes	ID

Description

The ENDPOINTS table contains all of the resource IDs that are tied to the different resource types. (Prior to Sun Role Manager 5.0, resources were called *endpoints*, and resource types were called *namespaces*.) A resource type can have multiple 'instances' and each instance has a corresponding resource. The NAMESPACE_ID field is used to reference the NAMESPACES table, and the field ID is unique.

Primary Keys

1.PK_ENDPOINTS - primary key on column ID

Foreign Keys

None

Indexes

None

top

ENDPOINT_POLICIES Table

Structure

	ENDPOINT_POLICIES
PK	ENDPOINT_ID
PK	POLICY_ID
PK	POLICY_VERSION_ID

Indexes

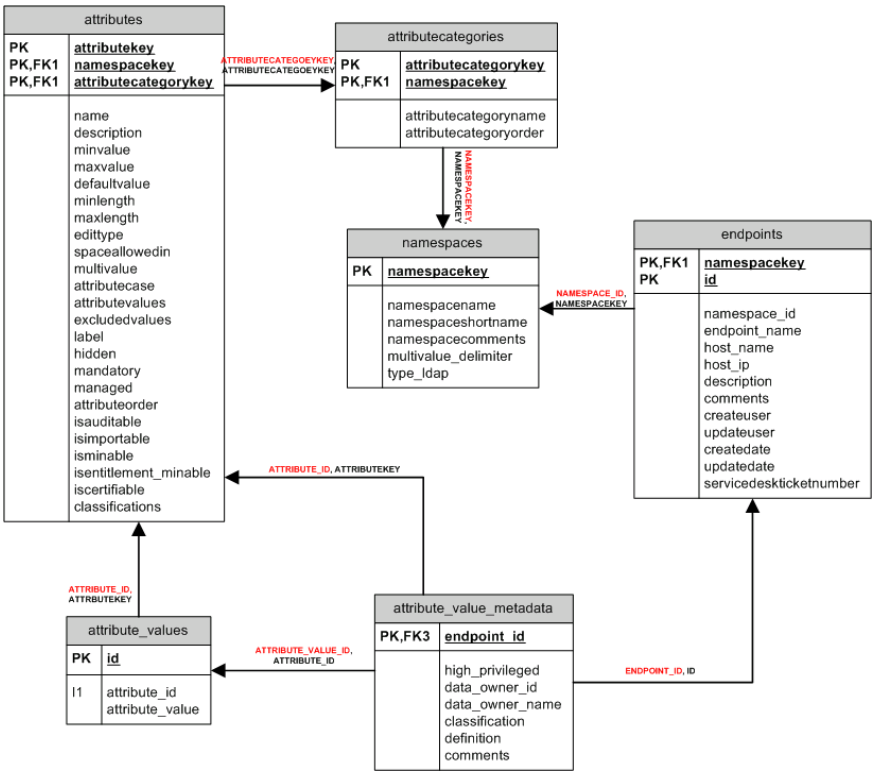
Index	PK	Unique	Keys
PK_ENDPOINT_POLICIES	✓	Yes	ENDPOINT_ID, POLICY_ID, POLICY_VERSION_ID

Description

Each policy can be associated with a resource (endpoint), and this is represented in a derived table, ENDPOINT_POLICIES. This table in turn references the tables ENDPOINTS and POLICIES through the respective primary keys.

top

Namespace Metadata Module Entity Relationship



top

Accounts Module

Tables in the Accounts Module

This chapter describes the tables that make up the Role Manager Accounts module.

ACCOUNTS Table

Structure

	ACCOUNTS
PK	ACCOUNTKEY
PK	NAMESPACEKEY
I1	NAME
	ID
	DESCRIPTION
	DOMAIN
	COMMENTS
	SUSPENDED
	LOCKED
	CREATEUSER
	UPDATEUSER
	CREATEDATE
	UPDATEDATE
	ATTR_VALUES_HASH
	ENDPOINT_ID
	ENDPOINT_NAME
	LAST_SYNCHRONIZED
	ACCOUNTTYPE_ID

Index

Index	PK	Unique	Keys
PK_ACCOUNTS	✓	Yes	ACCOUNTKEY, NAMESPACEKEY
IX_ACCOUNTS_ACCOUNTKEY		Yes	ACCOUNTKEY

Description

The ACCOUNTS table carries the accounts imported from the various Resource Types. Every account carries the Domain Name, Endpoint, and Status of the account. Also for audit purposes information about the account's Creation Date and Update Date, as well as the User who created the account and the user under which the account was updated is also saved in Role Manager. Each time the base accounts are updated, the ATTR_VALUES_HASH column is updated to keep track of the 'changed' accounts from the new feed. The account is also listed under its particular Resource Types by associating with the Namespace table using the NAMESPACEKEY as the reference.

Primary Keys

1. PK_ACCOUNTS - composite key on columns ACCOUNTKEY and NAMESPACEKEY

Foreign Keys

None

Indexes

1. IX_ACCOUNTS_ACCOUNTKEY - non unique index on column ACCOUNTKEY

top

ACCOUNT_TYPES Table

Structure

	ACCOUNT_TYPES
PK	ID
	LABEL DESCRIPTION

Indexes

Index	PK	Unique	Keys
PK_ACCOUNT_TYPES	✓	Yes	ID

Description

The different account types imported into Role Manager are listed under the 'ACCOUNT_TYPES' table. The following out-of-the-box account types are supported:

1. Provisioning account
2. High Privileged
3. Service Account
4. System Account

By default, all account types are set to 'Provisioning account' if not explicitly specified in the import process.

Primary Keys

1. PK_ACCOUNT_TYPES on column ID

Foreign Keys

None

Indexes

None

top

ACCOUNT_ATTRIBUTES Table

Structure

	ACCOUNT_ATTRIBUTES
PK	ID
I1 I2	ACCOUNT_ID PARENT_ID ACCT_ATTR_HIER_ID UPDATE_DATE

Indexes

Index	PK	Unique	Keys
PK_ACCOUNT_ATTRIBUTES_T	✓	Yes	ID
IX_ACCOUNT_ATTRIBUTES_ACID		No	ACCOUNT_ID
IX_ACCOUNT_ATTRIBUTES_PID		No	PARENT_ID

Description

The ACCOUNTS table carries only the account name and the audit values for that particular account. The ACCOUNT_ATTRIBUTES table carries references to the various parent accounts, along with hierarchy information within an account. It references the ACCOUNTS and the ACCT_ATTR_HIER_NODES tables by using the columns ACCOUNT_ID and ACCT_ATTR_HIER_ID respectively. These references help pull up information from the accounts and reference them to a particular account parent-child relationship.

Primary Keys

1.PK_ACCOUNT_ATTRIBUTES_T - primary key on column ID

Foreign Keys

None

Indexes

1.IX_ACCOUNT_ATTRIBUTES_ACID - non-unique index on column ACCOUNT_ID
2.IX_ACCOUNT_ATTRIBUTES_PID - non-unique index on column PARENT_ID

top

ARC_ACCOUNT_ATTRIBUTES Table

Structure

	ARC_ACCOUNT_ATTRIBUTES
	ID
I1	ACCOUNT_ID
I2	PARENT_ID
	ACCT_ATTR_HIER_ID
	UPDATE_DATE
	ARCHIVE_DATE

Index

Index	PK	Unique	Keys
IX_ARC_ACCT_ATTRIBUTES_ACID		No	ACCOUNT_ID
IX_ARC_ACT_ATTRIBUTES_PID		No	PARENT_ID

Description

The ARC_ACCOUNT_ATTRIBUTES table archives 'old/changed' account_attribute data from the ACCOUNT_ATTRIBUTES table whenever an account is changed/updated.

Primary Keys

None

Foreign Keys

None

Indexes

1.IX_ARC_ACCT_ATTRIBUTES_ACID - non-unique index on column ACCOUNT_ID
2.IX_ARC_ACCT_ATTRIBUTES_PID - non-unique index on column PARENT_ID

top

ACCT_ATTR_HIER_NODES Table

Structure

	ACCT_ATTR_HIER_NODES
PK	ID
I3	ROOT_ID
I2	PARENT_ID
I1	ATTRIBUTE_VALUE_ID
	LFT
	RGT
	UPDATE_DATE

Index

Index	PK	Unique	Keys
PK_ACCT_ATTR_HIER_NODES	✓	Yes	ID
IX_ACT_ATT_HIER_AV_ID		No	ATTRIBUTE_VALUE_ID
IX_ACT_ATT_HIER_RID		No	ROOT_ID
IX_ACT_ATT_HIER_PID		No	PARENT_ID

Description

The ACCT_ATTR_HIER_NODES maintains hierarchy information between attributes of an account. It has a unique ID, a ROOT_ID, and a PARENT_ID representing the root nodes and parent nodes respectively in the hierarchy. The ATTRIBUTE_VALUE_ID is a reference key to the ATTRIBUTE_VALUES table that contains the actual values of the attributes in the accounts. UPDATE_DATE stores information in case of an account modification. The fields LFT and RGT are used to maintain hierarchy information as well.

Primary Keys

1.PK_ACCT_ATTR_HIER_NODES - primary key on column ID

Foreign Keys

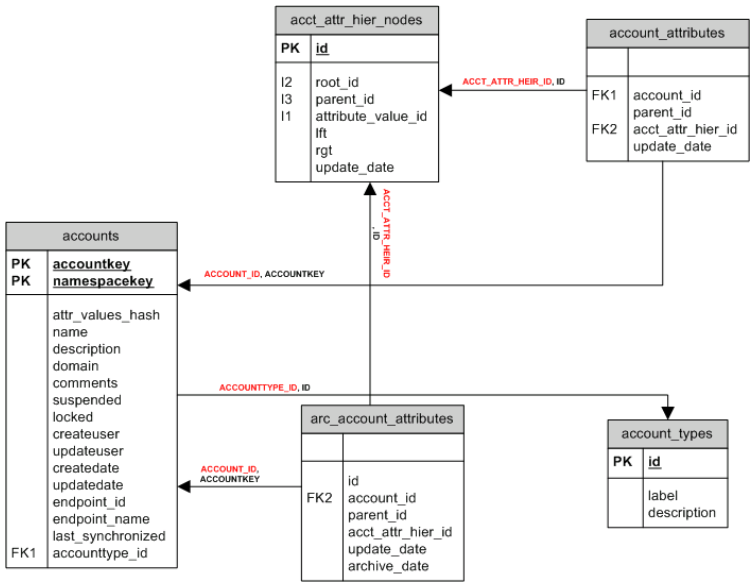
None

Indexes

1.IX_ACT_ATT_HIER_AV_ID - non-unique index on column ATTRIBUTE_VALUE_ID
2.IX_ACT_ATTR_HIER_RID - non-unique index on column ROOT_ID
3.IX_ACT_ATTR_HIER_PID - non-unique index on column PARENT_ID

top

Accounts Module Entity Relationship



top

Roles Module

Tables in the Roles Module

This chapter describes the tables that make up the Role Manager Roles module.

ROLES Table

Structure

	ROLES
PK	ROLEKEY
	ROLENAME
	ROLEDESCRIPTION
	ROLECOMMENTS
	DEPARTMENT
	STARTDATE
	ENDDATE
	CUSTOMPROPERTY1
	CUSTOMPROPERTY2
	CUSTOMPROPERTY3
	CUSTOMPROPERTY4
	CUSTOMPROPERTY5
	CUSTOMPROPERTY6
	CUSTOMPROPERTY7
	CUSTOMPROPERTY8
	CUSTOMPROPERTY9
	CUSTOMPROPERTY10
	CREATEUSER
	UPDATEUSER
	CREATEDATE
	UPDATEDATE
	JOBCODE
	SERVICEDESKTICKETNUMBER
	STATUSKEY
	HIGHPRIVELEGED
	BUSINESSAPPROVER
	TECHNICALAPPROVER
	USERASSOCIATIONBUAPPROVER
	USERASSOCIATIONTECHAPPROVER
	TYPE_ID
	WORKFLOW_ID
	CURRENT_VERSION_ID
	VERSION_COUNT
	LAST_SYNCHRONIZED
	DELEGABLE

Indexes

Index	PK	Unique	Keys
PK_ROLES	✔		ROLEKEY

Description

The roles that are designed in Role Manager are similar to the job functions in an organization. Each user can be associated with a role and assigned access levels. Roles are defined in the ROLES table and a unique Rolekey identifies each role. A role is assigned a ROLENAME, a ROLEDESCRIPTION, and a ROLECOMMENTS property. In addition, you can specify the department that a role is assigned to by populating the DEPARTMENT field. There are also 10 custom fields that can be used to add additional information about a role. For audit purposes, a role's CREATEUSER, UPDATEUSER, CREATEDATE, and UPDATEDATE fields are provided. The BUSINESSAPPROVER, TECHNICALAPPROVER, USERASSOCIATIONBUAPPROVER, and USERASSOCIATIONTECHAPPROVER fields are provided for Role Life Cycle Management Workflow process.

Primary Keys

1.PK_ROLES - primary key on column ROLEKEY

Foreign Keys

None

Indexes

None

top

ROLEHIERARCHY Table

Structure

	ROLEHIERARCHY
PK	ROLEKEY
PK	PARENTROLEKEY
PK	ROLE_VERSION_ID

Indexes

Index	PK	Unique	Keys
PK_ROLEHIERARCHY	✔		ROLEKEY, PARENTROLEKEY, ROLE_VERSION_ID

Description

Similar to Business Units, a hierarchy can exist between roles and these can be represented using the ROLEHIERARCHY table in Role Manager. Each role is associated with its parent in the PARENTROLEKEY field. This field contains the rolekey of the parent role.

Primary Keys

1.PK_ROLEHIERARCHY - composite primary key on columns ROLEKEY, PARENTROLEKEY and ROLE_VERSION_ID

Foreign Keys

None

Indexes

None

top

ROLE_VERSIONS Table

Structure

	ROLE_VERSIONS
PK	ID
I1	ROLE_ID
I2	VERSION_NUMBER
	ROLENAME
	ROLEDESCRIPTION
	ROLECOMMENTS
	DEPARTMENT
	STARTDATE
	END DATE
	CUSTOMPROPERTY1
	CUSTOMPROPERTY2
	CUSTOMPROPERTY3
	CUSTOMPROPERTY4
	CUSTOMPROPERTY5
	CUSTOMPROPERTY6
	CUSTOMPROPERTY7
	CUSTOMPROPERTY8
	CUSTOMPROPERTY9
	CUSTOMPROPERTY10
	CREATEUSER
	UPDATEUSER
	CREATEDATE
	UPDATEDATE
	JOBCODE
	SERVICEDESKTICKETNUMBER
	STATUSKEY
	HIGHPRIVILEGED
	BUSINESSAPPROVER
	TECHNICALAPPROVER
	USERASSOCIATIONBUAPPROVER
	USERASSOCIATIONTECHAPPROVER
	TYPE_ID
	WORKFLOW_ID
	VERSION_STATUS_ID
	BASE_VERSION_ID
	VERSION_COUNT
	LAST_APPROVER
	LAST_APPROVAL_DATE
	LAST_APPROVER_COMMENTS
	LAST_SYNCHRONIZED

Indexes

Index	PK	Unique	Keys
PK_ROLE_VERSIONS	✔		ID
IX_ROLE_VERSIONS			ROLE_ID,VERSION_NUMBER

Description

When a policy is added to or removed from a role in Role Manager, a role approval process is initiated by the System. A snapshot of the existing role is captured and saved into the ROLE_VERSIONS table prior to the initiation of the role approval process. This maintains the role's history, which can be accessed to compare previous versions.

Note - A new version of a role is not created when changes are made to the membership of the role (for example, if users are added or removed from the role).

Primary Keys

1.PK_ROLE_VERSIONS - composite primary key on columns ID

Foreign Keys

None

Indexes

1.IX_ROLE_VERSIONS - non-unique index on column ROLE_ID, VERSION_NUMBER

top

ROLESTATUSES Table

Structure

	ROLESTATUSES
PK	**STATUSKEY**
	LABEL

Indexes

Index	PK	Unique	Keys
PK_ROLESTATUSES	✔		STATUSKEY

Description

Similar to the Business Units, each role has a status that is stored in the ROLESTATUSES table. In Role Manager a role can have the following statuses depending on its stage in the Role Life Cycle:

- Active
- Inactive
- Composing
- Pending Approval
- Decommissioned

The STATUSKEY field is used to reference the ROLES table and the LABEL describes the status of the role.

Primary Keys

1.PK_ROLESTATUSES - primary key on column STATUSKEY

Foreign Keys

None

Indexes

None

top

ROLE_TYPES Table

Structure

	ROLE_TYPES
PK	**ID**
	LABEL DESCRIPTION

Indexes

Index	PK	Unique	Keys
PK_ROLE_TYPES	✔		ID

Description

The ROLE_TYPES table is used to define a Role category. The available Role category or Role Types are -

- Provisioning Role
- Access Control Role
- Organizational Role

Primary Keys

1. PK_ROLE_TYPES - composite primary key on columns ID

Foreign Keys

None

Indexes

None

top

ROLE_OWNERS Table

Structure

	ROLE_OWNERS
PK PK	**ID** **ROLE VERSION ID**
	ROLE_ID OWNER_ID OWNER_TYPE

Indexes

Index	PK	Unique	Keys
PK_ROLE_OWNERS	✔		ID, ROLE_VERSION_ID

Description

This ROLE_OWNERS table defines the relationship between a role and its owner which can be either a Globaluser or a role in Role Manager. ROLE_ID refers to the role owned in the ROLES table, whereas OWNER_ID refers to the GLOBALUSERS table.

Primary Keys

1.PK_ROLE_OWNERS - composite primary key on columns ID and ROLE_VERSION_ID

Foreign Keys

None

Indexes

None

top

ROLE_EXCLUSIONROLES Table

Structure

	ROLE_EXCLUSIONROLES
PK	ROLEKEY
PK	EXCLUSIONROLEKEY
PK	ROLE_VERSION_ID
	EXCLUSIONCOMMENTS

Indexes

Index	PK	Unique	Keys
PK_ROLE_EXCLUSIONROLES	✔		ROLEKEY, EXCLUSIONROLEKEY, ROLE_VERSION_ID

Description

The ROLE_EXCLUSIONROLES table defines the segregation of duties (SOD) between roles. Each Role defined for SOD has its role key and the corresponding (SOD) role's rolekey defined in the EXCLUSIONROLEKEY column.

Primary Keys

1.PK_ROLE_EXCLUSIONROLES - composite primary key on columns ROLEKEY, EXCLUSIONROLEKEY and ROLE_VERSION_ID

Foreign Keys

None

top

ROLE_POLICIES Table

Structure

	ROLE_POLICIES
PK	ROLEKEY
PK	POLICYKEY
PK	ROLE_VERSION_ID
	REQUIRED POLICY_VERSION_ID

Indexes

Index	PK	Unique	Keys
PK_ROLE_POLICIES	✔	Yes	ROLEKEY, POLICYKEY, ROLE_VERSION_ID

Description

ROLE_POLICIES is the association table for roles and policies. The associations vary with the role and policy versions, thus the columns role_version_id and policy_version_id are also maintained in this table.

Primary Keys

1.PK_ROLE_POLICIES - composite primary key on columns ROLEKEY, POLICYKEY, ROLE_VERSION_ID

Foreign Keys

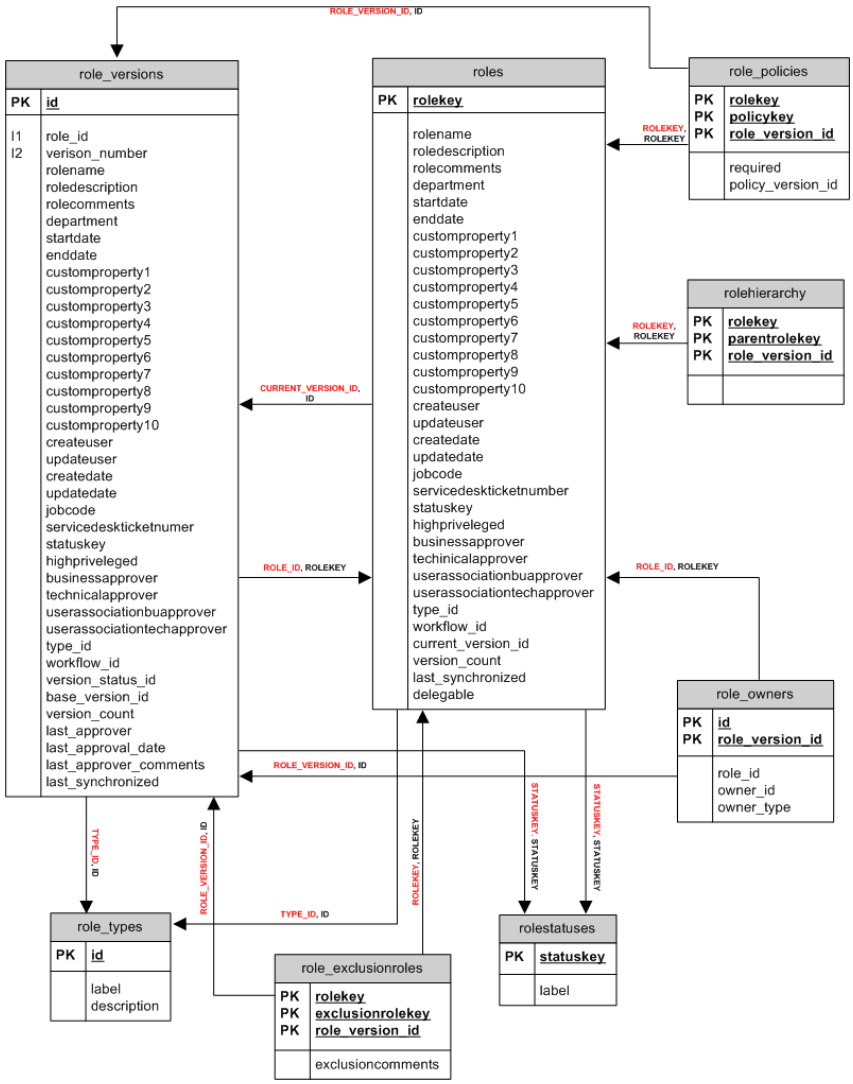
None

Indexes

None

top

Roles Module Entity Relationship



top

Policies Module

Tables in the Policies Module

This chapter describes the tables that make up the Role Manager Policies module.

POLICIES Table

Structure

	POLICIES
PK	POLICYKEY
PK	NAMESPACEKEY
	POLICYNAME
	POLICYCOMMENTS
	CREATEUSER
	UPDATEUSER
	CREATEDATE
	UPDATEDATE
	SERVICEDESKTICKETNUMBER
	TEMPLATE
	RISK_LEVEL
	CURRENT_VERSION_ID
	VERSION_COUNT
	STATUS_ID
	LAST_SYNCHRONIZED

Indexes

Index	PK	Unique	Keys
PK_POLICIES	✓	Yes	POLICYKEY, NAMESPACEKEY

Description

Each role is assigned one or more policies. The POLICES table contains information about these policies, which are identified by a unique POLICYKEY and a POLICYNAME. Each policy is associated with its resource type from the NAMESPACES table using the NAMESPACEKEY as a reference. For auditing purposes CREATEUSER, UPDATEUSER, CREATEDATE, and UPDATEDATE data is populated on every policy in this table.

Primary Keys

1.PK_POLICIES - composite primary key on columns POLICYKEY and NAMESPACEKEY

Foreign Keys

None

Indexes

None

top

POLICY_ATTR_HIER_NODES Table

Structure

	POLICY_ATTR_HIER_NODES
PK	ID
	ROOT_ID PARENT_ID ATTRIBUTE_VALUE_ID LFT RGT UPDATE_DATE

Index

Index	PK	Unique	Keys
PK_POLICY_ATTR_HIER_NODES	✓	Yes	ID
IX_POL_ATTR_HIER_AVID		No	ATTRIBUTE_VALUE_ID
IX_POL_ATTR_HIER_RID		No	ROOT_ID
IX_POL_ATTR_HIER_PID		No	PARENT_ID

Description

The POLICY_ATTR_HIER_NODES table maintains policy attribute hierarchy information. It has a unique ID, a ROOT_ID, and a PARENT_ID representing the root nodes and parent nodes respectively in the hierarchy. The ATTRIBUTE_VALUE_ID is a reference key to the ATTRIBUTE_VALUES table that contains the actual values of the attributes. UPDATE_DATE stores information in case of a modification to the account. Fields LFT and RGT are used to maintain hierarchy information as well.

Primary Key

1.PK_ID- primary key on column ID

Foreign Key

None

Indexes

None

top

POLICY_VERSIONS Table

Structure

	POLICY_VERSIONS
PK PK PK PK	ID VERSION_NUMBER POLICY_ID NAMESPACEKEY
	STATUS_ID POLICYNAME POLICYCOMMENTS CREATEUSER UPDATEUSER CREATEDATE UPDATEDATE SERVICEDESKTICKETNUMBER TEMPLATE RISK_LEVEL WORKFLOW_ID VERSION_STATUS_ID BASE_VERSION_ID VERSION_COUNT LAST_APPROVER LAST_APPROVAL_DATE LAST_APPROVER_COMMENTS LAST_SYNCHRONIZED

Indexes

Index	PK	Unique	Keys
PK_POLICY_VERSIONS	✓	Yes	ID, POLICY_ID, NAMESPACEKEY, VERSION_NUMBER

Description

Any modifications made to policies within Role Manager's Identity Warehouse module results in the creation of a new policy version. This policy version undergoes a Policy Approval Process. A snapshot of the existing policy is captured and saved in the POLICY_VERSIONS table prior to the initiation of the Policy Approval process.

Primary Keys

1.PK_POLICY_VERSIONS - composite primary key on column POLICY_ID, NAMESPACEKEY and VERSION_NUMBER

Foreign Keys

None

Indexes

None

top

POLICY_ATTRIBUTES Table

Structure

	POLICY_ATTRIBUTES
PK	ID
	POLICY_ID PARENT_ID POLICY_ATTR_HIER_ID RISK_LEVEL REQUIRED UPDATE_DATE POLICY_VERSION_ID

Indexes

Index	PK	Unique	Keys
PK_POLICY_ATTRIBUTES_T	✓	Yes	ID
IX_POLICY_ATTRIBUTES_PID		No	POLICY_ID
IX_POLICY_ATTRIBUTES_PRID		No	PARENT_ID

Description

Each policy has multiple attributes defined in it. (This is similar to the relationship between accounts and attributes.) The relationship between policies and attributes is defined in the POLICY_ATTRIBUTES table. The POLICY_ATTR_HIER_ID field references the ID of the POLICY_ATTR_HIER_NODES table, which links to the ATTRIBUTE_VALUES table.

Primary Keys

1. PK_POLICY_ATTRIBUTES_T - primary key on column ID

Foreign Keys

None

Indexes

1. IX_POLICY_ATTRIBUTES_PID - non-unique index on column POLICY_ID

2. IX_POLICY_ATTRIBUTES_PID - non-unique index on column PARENT_ID

top

POLICY_EXCLUSIONPOLICIES Table

Structure

	POLICY_EXCLUSIONPOLICIES
PK PK PK	POLICYKEY EXCLUSIONPOLICYKEY POLICY_VERSION_ID
	EXCLUSION COMMENTS

Indexes

Index	PK	Unique	Keys
PK_POLICY_EXCLUSIONPOLICIES	✓	Yes	POLICYKEY, POLICY_VERSION_ID, EXCLUSIONPOLICYKEY

Description

The POLICY_EXCLUSIONPOLICIES table defines the segregation of duties (SoD) between policies. Each policy defined for SoD has the policy's POLICYKEY and the corresponding SoD policy's key defined in the EXCLUSIONPOLICYKEY column.

Primary Keys

1.PK_POLICY_EXCLUSIONPOLICIES - composite primary key on columns POLICYKEY, POLICY_VERSION_ID and EXCLUSIONPOLICYKEY

Foreign Keys

None

Indexes

None

top

POLICY_OWNERS Table

Structure

	POLICY_OWNERS
PK	ID
	POLICY_ID OWNER_ID OWNER_TYPE POLICY_VERSION_ID

Indexes

Index	PK	Unique	Keys
PK_POLICY_OWNERS	✓	Yes	ID

Description

This table defines the relationship between a policy that is owned by a global user or a policy in Role Manager. POLICY_ID refers to the policy owned in the POLICIES table, whereas OWNER_ID refers to the GLOBALUSERS table.

Primary Keys

1.PK_POLICY_OWNERS - primary key on column ID

Foreign Keys

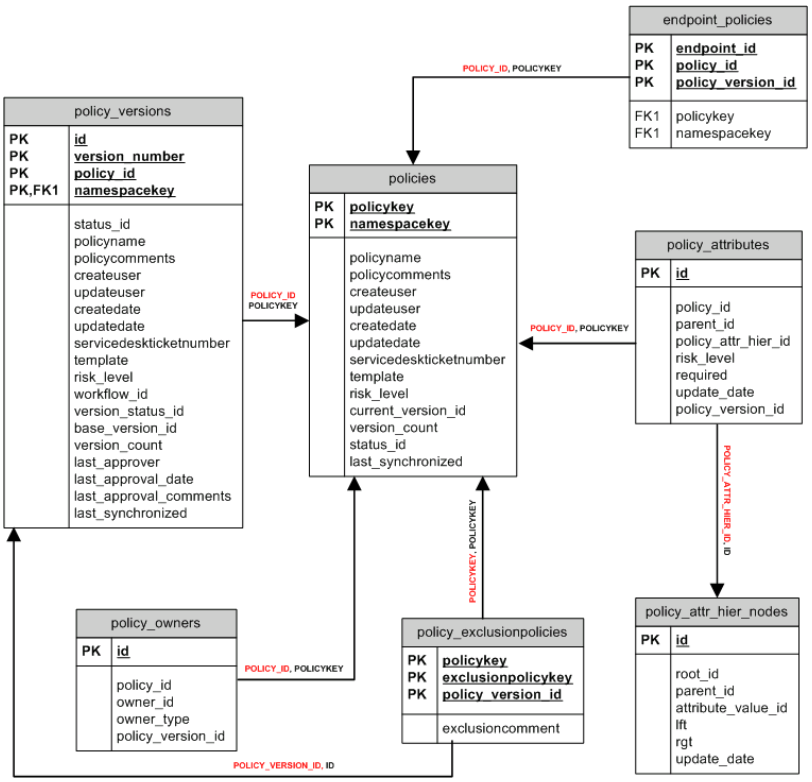
None

Indexes

None

top

Policies Module Entity Relationship



top

Applications Module

Tables in the Applications Module

This chapter describes the tables that make up the Role Manager Applications module.

APPLICATIONS Table

Structure

	APPLICATIONS
PK	ID
	APPLICATION_NAME APPLICATION_VERSION VENDOR ENVIRONMENT COMMENTS STATUS_ID DESCRIPTION UPDATE_DATE CREATE_DATE

Index

Index	PK	Unique	Keys
PK_APPLICATIONS	✓	Yes	ID

Description

The APPLICATIONS table contains general information about the applications in Role Manager.

Primary Keys

1.PK_APPLICATIONS - primary key on column ID

Foreign Keys

None

Indexes

None

top

APPLICATION_ATTRIBUTES Table

Structure

	APPLICATION_ATTRIBUTES
PK	APPLICATION_ID
PK	ENDPOINT_ID
PK	ATTRIBUTE_VALUE_ID

Index

Index	PK	Unique	Keys
PK_APP_ATTRIBUTES	✓	Yes	APPLICATION_ID, ENDPOINT_ID, ATTRIBUTE_VALUE_ID

Description

The APPLICATION_ATTRIBUTES table stores information about the endpoints (resources) and attribute values that define the conditions by which applications are linked to users. It references the ENDPOINTS table and the ATTRIBUTE_VALUES table by their respective IDs.

Primary Keys

1. PK_APP_ATTRIBUTES - composite primary key on columns APPLICATION_ID, ENDPOINT_ID, ATTRIBUTE_VALUE_ID

Foreign Keys

None

Indexes

None

top

APPLICATION_ACCOUNTS Table

Structure

	APPLICATION_ACCOUNTS
	APPLICATION_ID USER_ID ACCOUNT_ID

Index

None

Description

The APPLICATION_ACCOUNTS table saves information about user accounts, which is used to provide access to applications. This table references the APPLICATIONS, GLOBALUSERS, and ACCOUNTS tables by their corresponding IDs.

Primary Keys

None

Foreign Keys

None

Indexes

None

top

APPLICATION_OWNERS Table

Structure

	PK_APPLICATION_OWNERS
PK	ID
	APPLICATION_ID OWNER_ID OWNER_TYPE

Index

Index	PK	Unique	Keys
PK_APPLICATION_OWNERS	✓	Yes	ID

Description

The APPLICATION_OWNERS table contains information about the owner of every application. The OWNER_ID field contains the ID of the user (as found in the GLOBALUSERS table) who is the owner of the application. Every application-to-owner relationship is identified by a unique ID in this table.

Primary Keys

1.PK_APPLICATION_OWNERS - primary key on column ID

Foreign Keys

None

Indexes

None

top

APPLICATIONSTATUSES Table

Structure

	PK_APPLICATIONSTATUSES
PK	ID
	APPLICATION_ID OWNER_ID OWNER_TYPE

Index

Index	PK	Unique	Keys
PK_APPLICATIONSTATUSES	✓	Yes	ID

Description

The APPLICATIONSTATUSES table contains status labels of applications saved in the APPLICATIONS table.

Primary Keys

1.PK_APPLICATIONSTATUSES - primary key on column ID

Foreign Keys

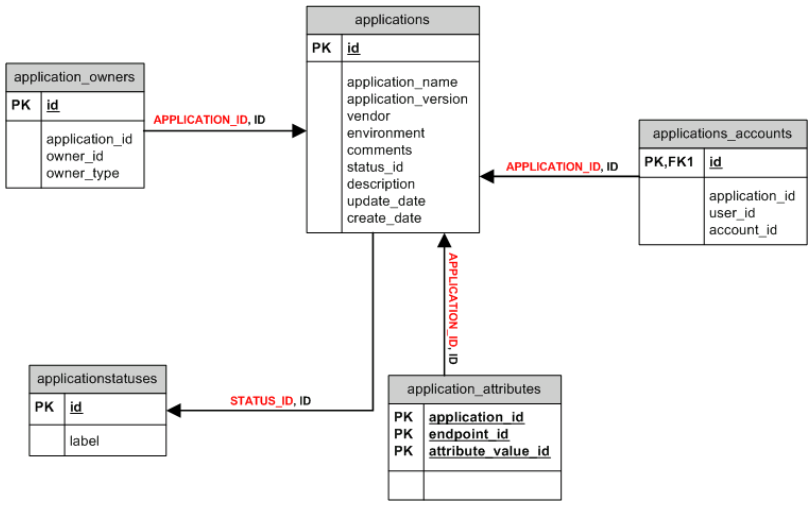
None

Indexes

None

top

Applications Module Entity Relationship



top

Import and Export Module

Tables in the Import Export Module

This chapter describes the tables that make up the Role Manager Import Export module.

IMPORT_RUNS Table

Structure

	IMPORT_RUNS
PK	IMPORT_ID
	SOURCE IMPORT_TYPE START_TIME END_TIME TOTAL_NUMBER_OF_RECORDS NUMBER_OF_RECORDS_IMPORTED NUMBER_OF_ERRORS DESCRIPTION IMPORTED_BY SUCCESS POST_IMPORT_TASKS

Index

Index	PK	Unique	Keys
PK_IMPORT_RUNS	✓		IMPORT_ID

Description

The IMPORT_RUNS table stores information about the Role Manager user interface while imports are taking place. It records the unique ID of the import and other valid details such as the SOURCE of the import, the IMPORT_TYPE, the START_TIME, the END_TIME, the number of records imported, and the number of accumulated errors.

Primary Keys

1.PK_IMPORT_RUNS - primary key on column IMPORT_ID

Foreign Keys

None

Indexes

None

top

IMPORT_RUN_STEPS Table

Structure

	IMPORT_RUN_STEPS
PK	ID
	IMPORT_RUN_ID NAME DESCRIPTION START_TIME END_TIME INPUT_NAME NUMBER_OF_INPUT_RECORDS OUTPUT_NAME NUMBER_OF_OUTPUT_RECORDS NUMBER_OF_ERRORS SUCCESS

Index

Index	PK	Unique	Keys
PK_IMPORT_RUN_STEPS	✓		ID

Description

Importing data into Role Manager may require several steps. Information about each import step is stored in the IMPORT_RUN_STEPS table, where the IMPORT_RUN_ID column refers to the IMPORT_ID column located in the IMPORT_RUNS table.

Primary Keys

1.PK_IMPORT_RUN_STEPS - primary key on column IMPORT_ID

Foreign Keys

None

Indexes

None

top

IMPORT_VALIDATION_EXCEPTIONS Table

Structure

	IMPORT_VALIDATION_EXCEPTIONS
	EXCEPTIONTYPE DESCRIPTION VALIDATIONDATE IMPORT_RUN_ID EXCEPTION_LEVEL IMPORT_RUN_STEP_ID

Description

Sun Role Manager can connect to a provisioning system and import accounts from that system. Any exceptions that occur during this process are stored in the IMPORT_VALIDATION_EXCEPTIONS table. The type and description of the exception are also stored in the table.

Primary Keys

None

Foreign Keys

None

Indexes

None

top

EXPORT_RUNS Table

Structure

	EXPORT_RUNS
PK	ID
	TARGET EXPORT_TYPE START_TIME END_TIME TOTAL_NUMBER_OF_RECORDS TOTAL_NUMBER_OF_RECORDS_EXPORTED NUMBER_OF_ERRORS DESCRIPTION EXPORTED_BY SUCCESS

Index

Index	PK	Unique	Keys
PK_EXPORT_RUNS	✓		ID

Description

While exports are taking place, the EXPORT_RUNS table records information about the Role Manager User Interface. It records the unique ID of the export process and other valid details, such as the type of export, the start and end times, the number of records exported, and the number of accumulated errors.

Primary Key

1. PK_EXPORT_RUNS - primary key on column ID

Foreign Key

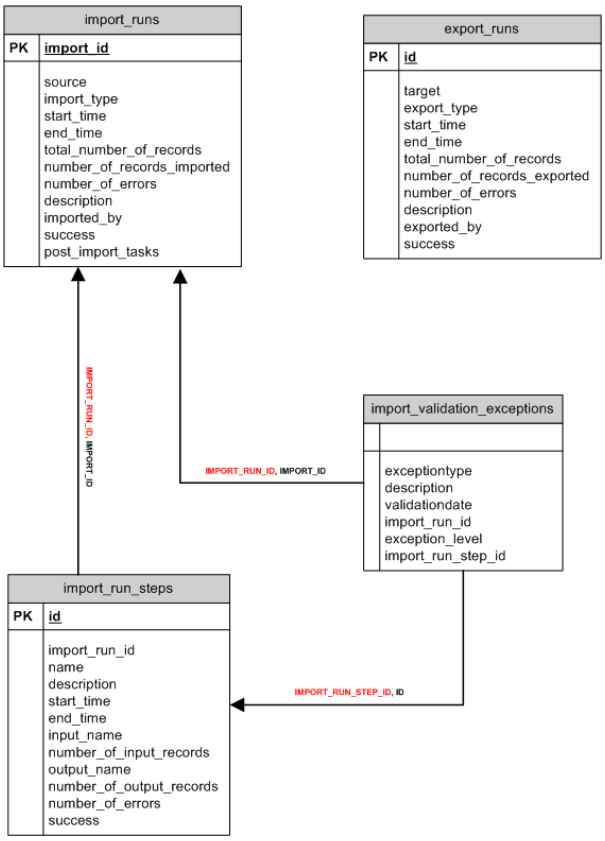
None

Indexes

None

top

Import Export Module Entity Relationship



top

Event Listener Module

Tables in the Event Listener Module

This chapter describes the tables that make up the Role Manager Event Listener module.

EVENT_LISTENERS Table

Structure

	REPORTS
PK	ID
	LISTENERNAME LISTENER_DESCRIPTION LISTENER_STATUS_ID LISTENER_CONDITION DATA_XML

Index

Index	PK	Unique	Keys
PK_EVENT_LISTENERS	✓	Yes	ID

Description

The EVENT_LISTENERS table is the primary table for the event listener module. It stores event listener data defined in Role Manager and identified by a unique ID. In addition, condition and action information is saved in the LISTENER_CONDITION and DATA_XML fields respectively.

Primary Keys

1.PK_EVENT_LISTENERS - primary key on column ID

Foreign Keys

None

Indexes

None

top

EVENT_TRIGGER_DATA Table

Structure

	REPORT_DEFINITIONS
PK	EVENTLISTENER_ID
PK	ACTION_TYPE
PK	OBJECTMODIFICATION_ID
PK	OBJECT_TYPE

Index

Index	PK	Unique	Keys
PK_EVENT_TRIGGER_DATA	✓	Yes	EVENTLISTENER_ID, ACTION_TYPE, OBJECTMODIFICATION_ID, OBJECT_TYPE

Description

The EVENT_TRIGGER_DATA table stores data about each action that is configured with a listener and defined in the EVENT_LISTENERS table. Details about the type of action and the object to be modified are stored in this table.

Primary Keys

1. PK_EVENT_TRIGGER_DATA - composite primary key on columns EVENTLISTENER_ID, ACTION_TYPE, OBJECTMODIFICATION_ID, OBJECT_TYPE

Foreign Keys

None

Indexes

None

top

OBJECT_MODIFICATIONS Table

Structure

	REPORT_DESIGNS
PK	ID
	OBJECT_ID OBJECT_TYPE DATA_XML

Index

Index	PK	Unique	Keys
PK_OBJECT_MODIFICATIONS	✓		ID

Description

The OBJECT_MODIFICATIONS table stores objects that have been modified in the system. Every object under modification is identified by a unique ID. Additional information about the object is stored in the DATA_XML column.

Primary Keys

PK_OBJECT_MODIFICATIONS - primary key on column ID

Foreign Keys

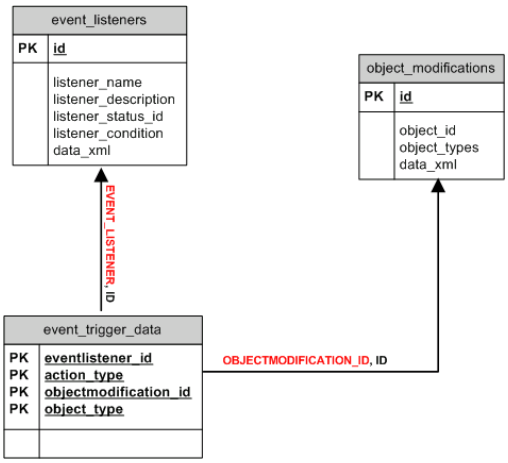
None

Indexes

None

top

Event Listener Module Entity Relationship



top

Identity Certification Module

Tables in the Identity Certification Module

This chapter describes the tables that make up the Role Manager Identity Certification module.

ID_CERTS Table

Structure

	ID_CERTS
PK	ID
	NAME PERIOD BUSINESSUNIT_ID STATE INCREMENTAL REPORT_ID CREATEUSER UPDATEUSER CREATEDATE UPDATEDATE TYPE END_DATE CERTIFIER_ID REMEDIATION_STATUS REMEDIATION_START_DATE REMEDIATION_END_DATE SUMMARY

Indexes

Index	PK	Unique	Keys
PK_ID_CERTS	✓	Yes	ID

Description

Each certification created in the Role Manager identity certification module is stored in the ID_CERTS table. A certification has a period of validity. The start date of the certification is saved in the PERIOD field, and the end date is stored in the END_DATE field. A business unit can also be associated with a certification and this information is saved in the table's BUSINESSUNIT_ID field. The remediation status of a supported certification can be tracked using the REMEDIATION_STATUS, REMEDIATION_START_DATE, and REMEDIATION_END_DATE fields. Finally, the following fields are provided for audit purposes: CREATEDATE, CREATEUSER, UPDATEDATE, and UPDATEUSER.

Primary Keys

1.PK_ID_CERTS - primary key on column ID

Foreign Keys

None

Indexes

None

top

IDC_CONFIGURATIONS Table

Structure

	IDC_CONFIGURATIONS
	IDC_ID CONFIGURATION_XML CREATEDATE CREATEUSER UPDATEDATE UPDATEUSER

Indexes

Index	PK	Unique	Keys

IX_IDC_ID		No	IDC_ID

Description

The IDC_CONFIGURATIONS table contains information about certification view configuration changes made by the administrator prior to launching the certification process. The CONFIGURATIONS_XML column holds each certification's configuration information in XML format. The CREATE_USER and CREATE_DATE fields store information about the user creating the certification and a timestamp marking when the certification was created.

Primary Keys

None

Foreign Keys

None

Indexes

1. IX_IDC_ID - non-unique index on column IDC_ID

top

IDC_USERS TABLE

Structure

	IDC_USERS
PK	ID
	IAM_USER_ID USERNAME FIRSTNAME LASTNAME MIDDLENAME STREET CITY STATEORPROVINCE ZIPORPOSTALCODE COUNTRYORREGION FAX PHONE EXTENSION MOBILE PAGER TITLE PRIMARYEMAIL SECONDARYEMAIL OFFICENAME DESCRIPTION COMMENTS SUSPENDEDDATE ENABLEDDATE DISABLEDDATE DELETEDDATE USERDATA EMPLOYEEID CUSTOMPROPERTY1 CUSTOMPROPERTY2 CUSTOMPROPERTY3 CUSTOMPROPERTY4 CUSTOMPROPERTY5 CUSTOMPROPERTY6 CUSTOMPROPERTY7 CUSTOMPROPERTY8 CUSTOMPROPERTY9 CUSTOMPROPERTY10 CUSTOMPROPERTY11 CUSTOMPROPERTY12 CUSTOMPROPERTY13 CUSTOMPROPERTY14 CUSTOMPROPERTY15 CUSTOMPROPERTY16 CUSTOMPROPERTY17 CUSTOMPROPERTY18 CUSTOMPROPERTY19 CUSTOMPROPERTY20 CREATEUSER UPDATEUSER CREATEDATE UPDATEDATE EMPLOYEETYPE SERVICEDESKTICKETNUMBER STARTDATE ENDDATE MANAGER BUSINESSAPPROVER TECHNICALAPPROVER DELEGATE LOCATION JOBCODES

Indexes

Index	PK	Unique	Keys
PK_IDC_USERS	✓	Yes	ID

Description

For entitlement certifications, users who require certification are defined in the IDC_USERS table. Each user record has a unique ID, a username, a firstname, a lastname, and a middlename. There are also fields that can store each user's street address, email address, and phone number. Custom fields are provided to capture special information. Because the user can be associated with a workflow, it is possible to save the user's manager, businessapprover, and technicalapprover. A delegate field is also present that a user can use to specify a delegated user. A statuskey specifies the user's status. For audit purposes the table includes fields for capturing the users's Creation Date and Update Date.

Primary Keys

1.PK_IDC_USERS - primary key on column ID

Foreign Keys

None

Indexes

None

top

ID_CERT_USERS Table

Structure

	ID_CERT_USERS
PK	**CERT_ID**
PK	**USER_ID**
	COMMENTS WORKS_FOR_ME REPORTS_TO CERTIFIED_BY

Indexes

Index	PK	Unique	Keys
PK_ID_CERT_USERS	✓	Yes	CERT_ID, USER_ID

Description

Every user who is certified in the certification process appears in the ID_CERT_USERS table. This table includes a CERT_ID column that maps to the ID field of the ID_CERTS table. A USER_ID is also assigned to the user. The REPORTS_TO field associates a user with the user's manager, and the CERTIFIED_BY field captures the username of the person updating the certification.

Primary Keys

1.PK_ID_CERT_USERS - composite primary key on columns CERT_ID and USER_ID

Foreign Keys

None

Indexes

None

top

IDC_ACCOUNTS Table

Structure

	IDC_ACCOUNTS
PK	**ID**
	IAM_ACCOUNT_ID NAMESPACEKEY NAME DESCRIPTION DOMAIN COMMENTS SUSPENDED LOCKED CREATEUSER UPDATEUSER CREATEDATE UPDATEDATE ENDPOINT_ID ENDPOINT_NAME ACCOUNTTYPE_ID

Index

Index	PK	Unique	Keys
PK_IDC_ACCOUNTS	✓	Yes	ID

Description

Each account for a user in the certification process is listed under the IDC_ACCOUNTS table. Each account is also associated with the IACCOUNTS and NAMESPACES tables by way of their respective reference keys. The table also stores the following information: account name, endpoint (resource), description, domain, and account status. ACCOUNTTYPE_ID correlates an account type with the certified account.

Primary Keys

1.PK_IDC_ACCOUNTS - primary key on column ID

Foreign Keys

None

Indexes

None

top

IDC_ACCOUNT_ATTRIBUTES Table

Structure

	IDC_ACCOUNT_ATTRIBUTES
PK	**ID**
	PARENT_ID ACCOUNT_ID ATTRIBUTE_ID ATTRIBUTE_VALUE HIGH_PRIVILEGED

UPDATEDATE

Indexes

Index	PK	Unique	Keys
PK_IDC_ACCOUNT_ATTRIBUTES	✓		ID

Description

The IDC_ACCOUNT_ATTRIBUTES table saves information about the value of the attribute in a certification. Each account in Role Manager has a list of attributes that need to be certified. These attributes are referenced from the ATTRIBUTES table, and the accounts are referenced from the ACCOUNTS table. The attribute value is also stored in this table.

Primary Keys

1.PK_IDC_ACCOUNT_ATTRIBUTES - primary key on column ID

Foreign Keys

None

Indexes

None

top

IDC_ATTR_VALUES Table

Structure

	IDC_ATTR_VALUES TABLE
PK	**ID**
	NAMESPACE_ID ENDPOINT_ID ATTRIBUTE_ID VALUE IAM_ATTR_VAL_ID CREATUSER UPDATEUSER CREATEDATE UPDATEDATE

Indexes

Index	PK	Unique	Keys
PK_IDC_ATTR_VALUES	✓	Yes	ID

Description

For Data Owner certifications, a list of attribute values can be specified for the certifications that needs to be performed. These attribute values are defined in the IDC_ATTR_VALUES table. Each value has a unique ID field, along with an ATTRIBUTE_ID field, an ENDPOINT_ID field, and a NAMESPACE_ID field that refer to the respective tables in the Identity Warehouse. **Note -** In Role Manager, the terminology "Namespace" and "Resource Type" mean the same thing.

Primary Keys

1. PK_IDC_ATTR_VALUES - primary key on column ID

Foreign Keys

None

Indexes

None

top

IDC_USER_ACCOUNTS Table

Structure

	IDC_USER_ACCOUNTS
PK	**CERT_ID**
PK	**USER_ID**
PK	**ACCOUNT_ID**
	CERTIFIED COMMENTS CERTIFIED_BY CERTIFICATION_DATE STATUS_END_DATE REMEDIATION_STATUS REMEDIATION_DATE REMEDIATION_COMMENTS

Indexes

Index	PK	Unique	Keys
PK_IDC_USER_ACCOUNTS	✓	Yes	CERT_ID, USER_ID, ACCOUNT_ID

Description

The IDC_USER_ACCOUNTS table is a derived table that associates user accounts with users who are subject to certification. This table includes the CERT_ID, the USER_ID, and the ACCOUNT_ID fields that reference the IDC_CERT_USERS and IDC_ACCOUNTS tables. Also associated with the IDC_USER_ACCOUNTS table is a certified flag that tracks if the account has been certified, as well as a CERTIFICATION_DATE field that stores the date/time of the certification, thus allowing for updates. In addition, the CERTIFIED_BY field contains the username of the person performing updates on the certification, which is required for auditing purposes. Remediation details of the certification process can be tracked using the REMEDIATION_STATUS, REMEDIATION_DATE, and REMEDIATION_COMMENTS fields.

Primary Keys

1.PK_IDC_USER_ACCOUNTS - composite primary key on columns CERT_ID, USER_ID and ACCOUNT_ID

Foreign Keys

None

Indexes

None

top

IDC_USER_ACCT_ATTRS Table

Structure

	IDC_USER_ACCT_ATTRS
PK	**CERT_ID**
PK	**USER_ID**
PK	**ACCOUNT_ID**
PK	**ACCOUNT_ATTRIBUTE_ID**
	CERTIFIED
	COMMENTS
	CERTIFIED_BY
	CHANGED_SINCE_LAST
	CERTIFICATION_DATE
	STATUS_END_DATE
	REMEDIATION_STATUS
	REMEDIATION_DATE
	REMEDIATION_COMMENTS

Indexes

Index	PK	Unique	Keys
PK_USER_IDC_ACCT_ATTRS	✓	Yes	CERT_ID, USER_ID, ACCOUNT_ID, ACCOUNT_ATTRIBUTE_ID

Description

The IDC_USER_ACCT_ATTRS table is a derived table that associates user account attributes with users in a certification. This table includes the CERT_ID, USER_ID, ACCOUNT_ID, and ACCOUNT_ATTRIBUTE_ID fields that reference the IDC_CERT_USERS, IDC_ACCOUNTS, and IDC_USER_ACCOUNTS tables. A certified flag tracks if accounts have been certified, and the CERTIFICATION_DATE field stores the date/time of the certification, thus allowing for updates. In addition, the CERTIFIED_BY field contains the username of the person that makes changes to the certification, which is required for auditing purposes. Remediation details of the certification process can be tracked using the REMEDIATION_STATUS, REMEDIATION_DATE, and REMEDIATION_COMMENTS fields.

Primary Keys

PK_USER_IDC_ACCT_ATTR - composite primary key on columns CERT_ID, USER_ID, ACCOUNT_ID, ACCOUNT_ATTRIBUTE_ID

Foreign Keys

None

Indexes

None

top

ID_ATTR_VAL_USERS Table

Structure

	ID_ATTR_VAL_USERS
PK	**CERT_ID**
PK	**USER_ID**
PK	**ATTR_VALUE_ID**
	NAMESPACE_ID
	ENDPOINT_ID
	ACCOUNT_ID
	ATTRIBUTE_ID
	CERTIFIED
	COMMENTS
	CERTIFIED_BY
	CERTIFICATION_DATE
	STATUS_END_DATE
	REMEDIATION_STATUS
	REMEDIATION_DATE
	REMEDIATION_COMMENTS
	ACCOUNT_NAME
	ENDPOINT_NAME
	NAMESPACE_NAME
	ATTRIBUTE_NAME

Indexes

Index	PK	Unique	Keys
PK_ID_ATTR_VAL_USERS	✓	Yes	CERT_ID, USER_ID, ATTR_VALUE_ID

Description

The ID_ATTR_VAL_USERS table is a derived table that associates users and attribute values with a certification. This table has CERT_ID, ATTR_VALUE_ID, and USER_ID fields that reference the ID_CERTS, IDC_ATTR_VALUES, and IDC_USERS tables. A certified flag records whether the association between the user and the attribute value has been certified, and the CERTIFICATION_DATE field stores the date/time of the certification, thus providing for updates. In addition, the CERTIFIED_BY field contains the username of the person who updated the certification, which is required for audit purposes. Remediation details of the certification process can be tracked using the REMEDIATION_STATUS, REMEDIATION_DATE, and REMEDIATION_COMMENTS fields.

Primary Keys

PK_ID_ATTR_VAL_USERS - composite primary key on columns CERT_ID, USER_ID, ATTR_VALUE_ID

Foreign Keys

None

Indexes

None

top

ID_CERT_ATTR_VALUES Table

Structure

	ID_CERT_ATTR_VALUES
PK	CERT_ID
PK	ATTR_VALUE_ID
	GLOSSARY_DEF
	COMMENTS
	TECH_DESCRIPTION
	HIGH_PRIVILEGED
	CERTIFIED
	BELONGS_TO_ME
	CERTIFIED_BY
	CERTIFICATION_DATE
	STATUS_END_DATE

Indexes

Index	PK	Unique	Keys
PK_ID_CERT_ATTR_VALUES	✓	Yes	CERT_ID, ATTR_VALUE_ID

Description

The ID_CERT_ATTR_VALUES table is a derived table that associates attribute values with a certification. This table has CERT_ID and ATTR_VALUE_ID fields that reference the ID_CERTS and IDC_ATTR_VALUES tables. A certified flag records if the association between the attribute value and the certification has been certified, and the CERTIFICATION_DATE field stores the date/time of the certification, thus providing for updates. In addition, the CERTIFIED_BY field contains the username of the person who updated the certification, which is required for audit purposes. Remediation details of the certification process can be tracked using the REMEDIATION_STATUS, REMEDIATION_DATE, and REMEDIATION_COMMENTS fields.

Primary Keys

PK_ID_CERT_ATTR_VALUES - composite primary key on columns CERT_ID, ATTR_VALUE_ID

Foreign Keys

None

Indexes

None

top

IDC_ROLES Table

Structure

	IDC_ROLES
PK	ID
	IAM_ROLE_ID
	ROLENAME
	ROLEDESCRIPTION
	ROLECOMMENTS
	DEPARTMENT
	STARTDATE
	ENDDATE
	CUSTOMPROPERTY1
	CUSTOMPROPERTY2
	CUSTOMPROPERTY3
	CUSTOMPROPERTY4
	CUSTOMPROPERTY5
	CUSTOMPROPERTY6
	CUSTOMPROPERTY7
	CUSTOMPROPERTY8
	CUSTOMPROPERTY9
	CUSTOMPROPERTY10
	HIGHPRIVELEGED
	JOBCODE
	SERVICEDESKTICKETNUMBER
	BUSINESSAPPROVER
	TECHNICALAPPROVER
	USERASSOCIATIONBUAPPROVER
	USERASSOCIATIONTECHAPPROVER
	CREATEUSER
	UPDATEUSER
	CREATEDATE
	UPDATEDATE

Indexes

Index	PK	Unique	Keys
PK_IDC_ROLES	✓	Yes	ID

Description

The IDC_ROLES table lists each role associated with the user in the certification process. Each role is associated with the ROLES table by way of the reference rolekey, and custom fields are provided to capture custom role information. For audit purposes, the table includes fields for capturing createuser and createtime information, and updateuser and updatetime information for a role. The BusinessApprover, TechnicalApprover, UserAssociationBusinessApprover, and UserAssociationTechnicalApprover fields are provided for the Workflow process. Each role is associated with an Identity Certificate through the ID field.

Primary Keys

1.PK_IDC_ROLES - primary key on column ID

Foreign Keys

None

Indexes

None

top

ID_CERT_ROLES Table

Structure

	ID_CERT_ROLES
PK	**CERT_ID**
PK	**ROLE_ID**
	CERTIFIED COMMENTS CERTIFIED_BY CERTIFICATION_DATE REMEDIATION_STATUS REMEDIATION_DATE REMEDIATION_COMMENTS

Indexes

Index	PK	Unique	Keys
PK_ID_CERT_ROLES	✓	Yes	CERT_ID, ROLE_ID

Description

The ID_CERT_ROLES table records every role that is certified in the certification process. The certification ID maps to the ID field in the ID_CERTS table, and a ROLE_ID is assigned to capture the role ID. A CERTIFIED flag records if the roles have been certified, and the CERTIFICATION_DATE field stores the date/time of the certification, thus providing for updates. In addition, the CERTIFIED_BY field contains the username of the person updating the certification, which is required for auditing purposes. Remediation details of the certification process can be tracked using the REMEDIATION_STATUS, REMEDIATION_DATE, and REMEDIATION_COMMENTS fields.

Primary Keys

1.PK_ID_CERT_ROLES - composite primary key on column CERT ID, ROLE_ID

Foreign Keys

None

Indexes

None

top

IDC_USER_ROLES Table

Structure

	IDC_USER_ROLES
PK	**CERT_ID**
PK	**USER_ID**
PK	**ROLE_ID**
	CERTIFIED COMMENTS CERTIFIED_BY CERTIFICATION_DATE REMEDIATION_STATUS REMEDIATION_DATE REMEDIATION_COMMENTS

Indexes

Index	PK	Unique	Keys
PK_IDC_USER_ROLES	✓	Yes	CERT_ID, USER_ID, ROLE_ID

Description

The IDC_USER_ROLES table is a composite table that associates roles with user certifications. The USER_ID, ROLE_ID, and CERT_ID fields reference the IDC_CERT_USERS and IDC_ROLES tables. A CERTIFIED flag records if the account has been certified, and the CERTIFICATION_DATE field stores the date/time of the certification, thus providing for updates. In addition, the CERTIFIED_BY field contains the username of the person updating the certification, which is required for auditing purposes. Remediation details of the certification process can be tracked using the REMEDIATION_STATUS, REMEDIATION_DATE, and REMEDIATION_COMMENTS fields.

Primary Keys

1.PK_IDC_USER_ROLES - composite primary key on columns USER_ID, ROLE_ID and CERT_ID

Foreign Keys

None

Indexes

None

top

IDC_POLICIES Table

Structure

	IDC_POLICIES
PK	**ID**
	IAM_POLICIES_ID NAMESPACE_ID ENDPOINT_ID POLICY_NAME DESCRIPTION COMMENTS CREATE_USER UPDATE_USER CREATE_DATE UPDATE_DATE

Indexes

Index	PK	Unique	Keys
PK_IDC_POLICIES	✓	Yes	ID

Description

The IDC_POLICIES table saves the policy information of a certification with roles. A policy is assigned a POLICY_NAME, a DESCRIPTION, and COMMENTS. An association with a Namespace (Resource Type) and an Endpoint (Resource) is made by utilizing NAMESPACE_ID and ENDPOINT_ID respectively.

Primary Keys

1.PK_IDC_POLICIES - composite primary key on columns ID

Foreign Keys

None

Indexes

None

top

IDC_POLICY_ATTRIBUTES Table

Structure

	IDC_POLICY_ATTRIBUTES
PK	ID
	PARENT_ID POLICY_ID ATTRIBUTE_ID ATTRIBUTE_VALUE HIGH_PRIVILEGED UPDATE_DATE

Indexes

Index	PK	Unique	Keys
PK_IDC_POLICY_ATTRIBUTES	✓	Yes	ID

Description

A snapshot of various policy attributes are captured at the moment of certification and saved in the IDC_POLICY_ATTRIBUTES table. These attributes are similar to the account attributes defined. The ATTRIBUTE_VALUE field holds the actual value of the policies, whereas the POLICIES and ATTRIBUTES tables are referenced by way of the POLICY_ID and ATTRIBUTE_ID fields. The PARENT_ID field associates the policy with its parent.

Primary Keys

1.PK_IDC_POLICY_ATTRIBUTES - primary key on column ID

Foreign Keys

None

Indexes

None

top

IDC_ROLE_POLICIES Table

Structure

	IDC_ROLE_POLICIES
PK PK PK	CERT_ID ROLE_ID POLICY_ID
	CERTIFIED COMMENTS CERTIFICATION_DATE STATUS_END_DATE REMEDIATION_STATUS REMEDIATION_DATE REMEDIATION_COMMENTS

Indexes

Index	PK	Unique	Keys
PK_IDC_ROLE_POLICIES	✓	Yes	CERT_ID, ROLE_ID, POLICY_ID

Description

The IDC_ROLE_POLICIES table is a derived table that associates roles and policies with certifications. The CERT_ID, ROLE_ID, and POLICY_ID fields reference the IDC_POLICIES, IDC_ROLES, and ID_CERT_ROLES tables. A CERTIFIED flag records if the role to policy association has been certified, and the CERTIFICATION_DATE field stores the date/time of the certification, thus providing for updates. In addition, the CERTIFIED_BY field contains the username of the person updating the certification, which is required for auditing purposes. Remediation details of the certification process can be tracked using the REMEDIATION_STATUS, REMEDIATION_DATE, and REMEDIATION_COMMENTS fields.

Primary Keys

1.PK_IDC_ROLE_POLICIES - composite primary key on column CERT_ID, ROLE_ID, POLICY_ID

Foreign Keys

None

Indexes

None

top

IDC_ROLE_POLICY_ATTRS Table

Structure

	IDC_ROLE_POLICY_ATTRS

PK	**CERT_ID**
PK	**ROLE_ID**
PK	**POLICY_ID**
PK	**POLICY_ATTRIBUTE_ID**
	CERTIFIED COMMENTS CERTIFIED_BY CHANGED_SINCE_LAST CERTIFICATION_DATE STATUS_END_DATE REMEDIATION_STATUS REMEDIATION_DATE REMEDIATION_COMMENTS

Indexes

Index	PK	Unique	Keys
PK_IDC_ROLE_POLICY_ATTRS	✓	Yes	CERT_ID, ROLE_ID, POLICY_ID, POLICY_ATTRIBUTE_ID

Description

The IDC_ROLE_POLICY_ATTR table is a derived table that associates roles, policies, and role and policy attributes with certifications. The CERT_ID, ROLE_ID, POLICY_ID, and POLICY_ATTRIBUTE_ID fields reference the following tables: IDC_POLICIES, IDC_ROLES, IDC_ROLES_POLICIES and ID_CERT_ROLES. A CERTIFIED flag records if the association between roles, policies, and role and policy attributes have been certified, and the CERTIFICATION_DATE field stores the date/time of the certification, thus providing for updates. In addition, the CERTIFIED_BY field contains the username of the person updating the certification, which is required for auditing purposes. Remediation details of the certification process can be tracked using the REMEDIATION_STATUS, REMEDIATION_DATE, and REMEDIATION_COMMENTS fields.

Primary Keys

1.PK_IDC_ROLE_POLICIES - composite primary key on column CERT_ID, ROLE_ID, POLICY_ID, POLICY_ATTRIBUTE_ID

Foreign Keys

None

Indexes

None

top

REMEDIATION_CONFIG Table

Structure

	REMEDIATION_CONFIG
PK	**ENDPOINT_ID**
	REMEDIATION_MODE REMEDIATION_STEPS IAM_CONNECTION_ID CREATEUSER UPDATEUSER CREATEDATE UPDATEDATE

Indexes

Index	PK	Unique	Keys
PK_REMEDIATION_CONFIG	✓	Yes	ENDPOINT_ID

Description

The REMEDIAITON_CONFIG table stores details that define how closed loop remediation of certification data should occur for each resource (endpoint). This process can happen either automatically or manually. Fields are provided to record the configured remediation mode (that is, auto or manual), the reference to the IAM connection (if the mode is auto), the textual description of the remediation steps (if the mode is manual), and the unique ID.

Primary Keys

1.PK_REMEDIATION_CONFIG - primary key on column ENDPOINT_ID

Foreign Keys

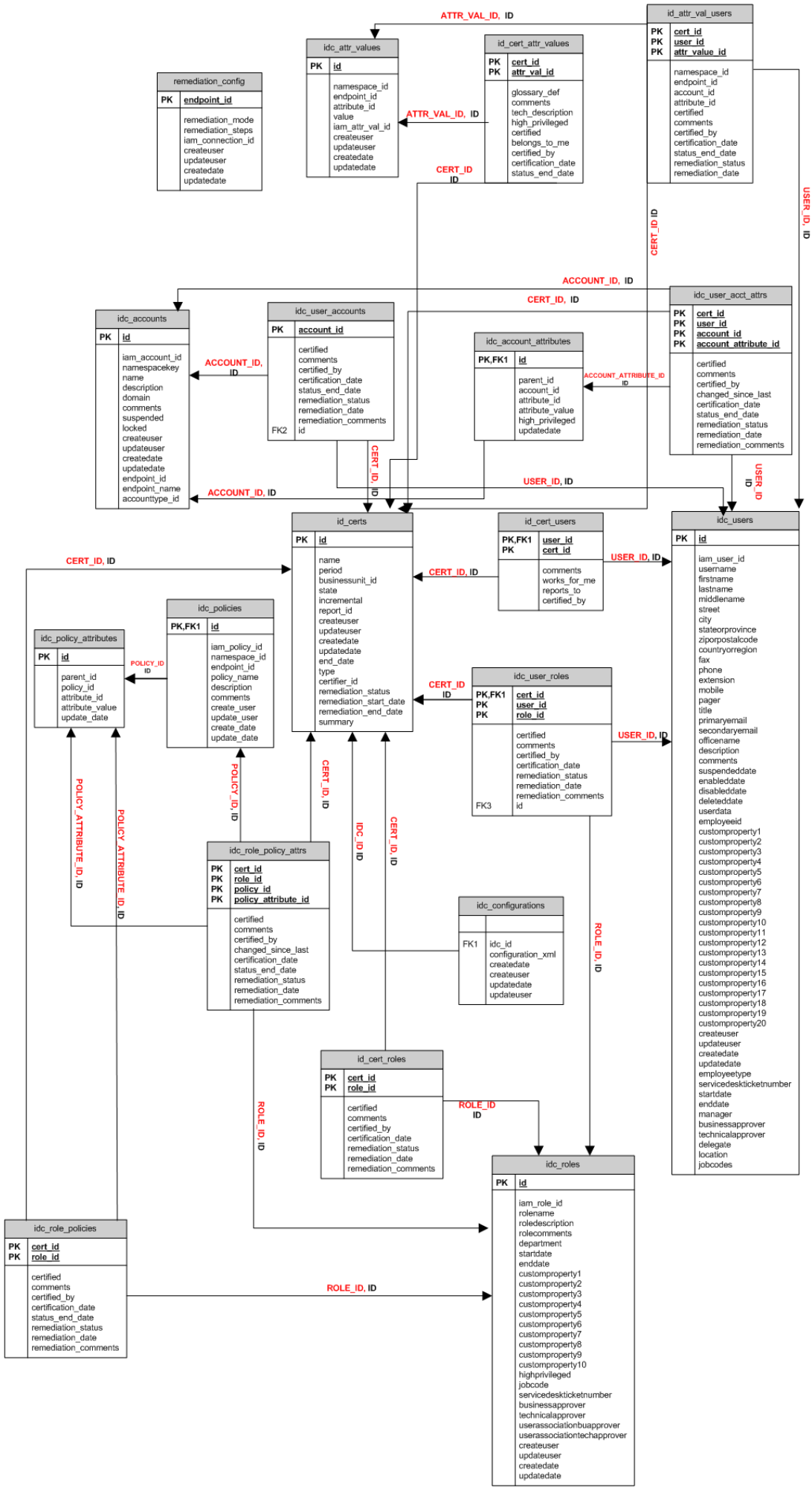
None

Indexes

None

top

Identity Certification Module Entity Relationship


id_attr_val_users
PK cert_id
PK user_id
PK attr_value_id
namespace_id
endpoint_id
account_id
attribute_id
certified
comments
certified_by
certification_date
status_end_date
remediation_status
remediation_date
idc_attr_values
PK id
namespace_id
endpoint_id
attribute_id
value
iam_attr_val_id
createuser
updateuser
createdate
updatedate
id_cert_attr_values
PK cert_id
PK attr_val_id
glossary_def
comments
tech_description
high_privileged
certified
belongs_to_me
certified_by
certification_date
status_end_date
remediation_config
PK endpoint_id
remediation_mode
remediation_steps
iam_connection_id
createuser
updateuser
createdate
updatedate
ATTR_VAL_ID, ID
CERT_ID ID
CERT_ID ID
USER_ID, ID
ACCOUNT_ID, ID
CERT_ID, ID
idc_accounts
PK id
iam_account_id
namespacekey
name
description
domain
comments
suspended
locked
createuser
updateuser
createdate
updatedate
endpoint_id
endpoint_name
accounttype_id
idc_user_accounts
PK account_id
certified
comments
certified_by
certification_date
status_end_date
remediation_status
remediation_date
remediation_comments
FK2 id
idc_account_attributes
PK,FK1 id
parent_id
account_id
attribute_id
attribute_value
high_privileged
updatedate
idc_user_acct_attrs
PK cert_id
PK user_id
PK account_id
PK account_attribute_id
certified
comments
certified_by
changed_since_last
certification_date
status_end_date
remediation_status
remediation_date
remediation_comments
ACCOUNT_ID, ID
ACCOUNT_ATTRIBUTE_ID ID
CERT_ID, ID
USER_ID, ID
ACCOUNT_ID, ID
USER_ID, ID
id_certs
PK id
name
period
businessunit_id
state
incremental
report_id
createuser
updateuser
createdate
updatedate
end_date
type
certifier_id
remediation_status
remediation_start_date
remediation_end_date
summary
id_cert_users
PK,FK1 user_id
PK cert_id
comments
works_for_me
reports_to
certified_by
idc_users
PK id
iam_user_id
username
firstname
lastname
middlename
street
city
stateorprovince
ziporpostalcode
countryorregion
fax
phone
extension
mobile
pager
title
primaryemail
secondaryemail
officename
description
comments
suspendeddate
enableddate
disableddate
deleteddate
userdata
employeeid
customproperty1
customproperty2
customproperty3
customproperty4
customproperty5
customproperty6
customproperty7
customproperty8
customproperty9
customproperty10
customproperty11
customproperty12
customproperty13
customproperty14
customproperty15
customproperty16
customproperty17
customproperty18
customproperty19
customproperty20
createuser
updateuser
createdate
updatedate
employeetype
servicedesktickettnumber
startdate
enddate
manager
businessapprover
technicalapprover
delegate
location
jobcodes
CERT_ID, ID
CERT_ID, ID
USER_ID, ID
idc_policies
PK,FK1 id
iam_policy_id
namespace_id
endpoint_id
policy_name
description
comments
create_user
update_user
create_date
update_date
idc_policy_attributes
PK id
parent_id
policy_id
attribute_id
attribute_value
update_date
POLICY_ID ID
idc_user_roles
PK,FK1 cert_id
PK user_id
PK role_id
certified
comments
certified_by
certification_date
remediation_status
remediation_date
remediation_comments
FK3 id
CERT_ID ID
USER_ID, ID
POLICY_ATTRIBUTE_ID, ID
POLICY_ATTRIBUTE_ID, ID
POLICY_ID, ID
CERT_ID, ID
IDC_ID ID
CERT_ID, ID
idc_role_policy_attrs
PK cert_id
PK role_id
PK policy_id
PK policy_attribute_id
certified
comments
certified_by
changed_since_last
certification_date
status_end_date
remediation_status
remediation_date
remediation_comments
idc_configurations
FK1 idc_id
configuration_xml
createdate
createuser
updatedate
updateuser
ROLE_ID, ID
id_cert_roles
PK cert_id
PK role_id
certified
comments
certified_by
certification_date
remediation_status
remediation_date
remediation_comments
ROLE_ID ID
ROLE_ID, ID
idc_roles
PK id
iam_role_id
rolename
roledescription
rolecomments
department
startdate
enddate
customproperty1
customproperty2
customproperty3
customproperty4
customproperty5
customproperty6
customproperty7
customproperty8
customproperty9
customproperty10
highprivileged
jobcode
servicedesktickettnumber
businessapprover
technicalapprover
userassociationbuapprover
userassociationtechapprover
createuser
updateuser
createdate
updatedate
idc_role_policies
PK cert_id
PK role_id
certified
comments
certified_by
certification_date
status_end_date
remediation_status
remediation_date
remediation_comments
ROLE_ID, ID

top

Identity Audit Module

Tables in the Identity Audit Module

This chapter describes the tables that make up the Identity Audit module.

IDA_POLICIES Table

Structure

	IDA_POLICIES
PK	ID
U1	RULESET_ID POLICY_NAME EXCEPTION_SEVERITY OWNER OWNER_ID REMEDIATOR REMEDIATOR_ID

Index

Index	PK	Unique	Keys
PK_IDA_POLICIES	✓	Yes	ID
IX_IDA_POLICY_NAMES		Yes	POLICY_NAME

Description

The Identity Audit (IDA) module identifies policy violations and the IDA_POLICES table stores a collection of rules defined by the RULE_SETS table. Every policy stored in this table has an OWNER and a REMEDIATOR.

Primary Keys

1.PK_IDA_POLICIES - primary key on column ID

Foreign Keys

None

Indexes

1.IX_IDA_POLICY_NAMES - unique index on column POLICY_NAME

top

IDA_POLICY_VIOLATIONS Table

Structure

	IDA_POLICY_VIOLATIONS
PK	ID
	IDA_POLICY_ID USER_ID LASTDATE_DETECTED DETECTION_COUNT CLOSE_DATE MITIGATION_EXPIRATION_DATE COMMENTS POLICY_VIOLATION_STATE SEVERITY ASSIGNED_TO_DATE ASSIGNED_TO ASSIGNED_TO_ID CURRENT_TRAIL_ID CREATEUSER UPDATEUSER CREATEDATE UPDATEDATE REMINDER_COUNT LAST_REMINDER_DATE CREATE_SCAN_ID

Index

Index	PK	Unique	Keys
PK_IDA_POLICY_VIOLATIONS	✓	Yes	ID

Description

The IDA_POLICY_VIOLATIONS table holds information about policy violations detected by an identity audit scan. Each entry has a unique ID and the following fields: IDA_POLICY_ID, USER_ID, LASTDATE_DETECTED, DETECTION_COUNT, POLICY_VIOLATION_STATE, SEVERITY (1-High, 2-Medium, 3-Low), CREATEDATE, and UPDATEDATE. The POLICY_VIOLATION_STATE stores the state of the violation in the form of an integer that can have the following values: 1-Open, 2-Closed, 3-Remediated, or 4-Mitigated.

Primary Keys

1.PK_IDA_POLICY_VIOLATIONS - primary key on column ID

Foreign Keys

None

Indexes

None

top

IDA_POLICY_VIOLATION_CAUSES Table

Structure

	IDA_POLICY_VIOLATION_CAUSES
PK	ID
	IDA_POLICY_VIOLATION_ID IDA_RULE_ID IDA_RULE_VERSION_ID RBACX_OBJECT_ID RBACX_OBJECT_TYPE DESCRIPTION STATUS

Index

Index	PK	Unique	Keys
PK_IDA_POLICY_VIOLATION_CAUSES	✓	Yes	ID

Description

The IDA_POLICY_VIOLATION_CAUSES table holds information about the causes behind policy violations as detected by an identity audit scan. Each entry has a unique ID. The IDA_POLICY_VIOLATION_ID field is derived from the IDA_POLICY_VIOLATION table. The IDA_RULE_ID and IDA_RULE_VERSION_ID fields contain information about the rule set that the policy was run against, whereas the RBACX_OBJECT_ID and RBACX_OBJECT_TYPE fields store information about the Role Manager object against which the policy violation was detected. Additional information, such as the DESCRIPTION of the policy violation cause and the policy violation STATUS are also stored in this table.

Primary Keys

1.PK_IDA_POLICY_VIOLATION_CAUSES - primary key on column ID

Foreign Keys

None

Indexes

None

top

IDA_SCANS Table

Structure

	IDA_SCANS
PK	ID
	NAME SCAN_TYPE STATUS_ID RESULT_APPLIED DESCRIPTION CREATE_USER CREATE_DATE UPDATE_DATE RUN_COMPLETE_DATE NEXT_RUN_DATE CRON_EXPRESSION USER_COUNT VIOLATION_COUNT CONFIGURATION_XML DATA_XML

Index

Index	PK	Unique	Keys
PK_IDA_SCANS	✓	Yes	ID

Description

The IDA_SCANS table saves information about the audit scans that are run in Role Manager. Every scan is identified by a unique ID from the IDA_SCANS table. This table also saves information like NAME, SCAN_TYPE, STATUS_ID, and DESCRIPTION. The RESULT_APPLIED field indicates if the scan was executed (run) or just previewed. When the scan is scheduled to run periodically, information is maintained in this table in the following fields: CREATE_USER, CREATE_DATE, UPDATE_DATE, RUN_COMPLETE_DATE, and NEXT_RUN_DATE. The USER_COUNT field contains the number of users that were scanned, whereas the VIOLATION_COUNT field contains the number of violations detected as a result of the scan run. The configuration information regarding each scan is saved as XML in the CONFIGURATION_XML field, and the scan data information is saved in the DATA_XML field.

Primary Key

PK_IDA_SCANS - primary key on column ID

Foreign Key

None

Indexes

None

top

IDA_POLICY_VIOLATIONS_TRAILS Table

Structure

	IDA_POLICY_VIOLATIONS_TRAILS
PK	ID
	IDA_POLICY_VIOLATIONS_ID COMMENTS POLICY_VIOLATION_STATE ASSIGNED_TO CREATEUSER CREATEDATE

Index

Index	PK	Unique	Keys

PK_IDA_POLICY_VIOLATIONS_TRAILS	✓	Yes	ID

Description

The IDA_POLICY_VIOLATIONS_TRAILS table stores the steps taken by an actor (that is, the Role Manager user) to resolve a policy violation detected during an identity audit scan.

Primary Keys

1.PK_IDA_POLICY_VIOLATIONS_TRAILS - primary key on column ID

Foreign Keys

None

Indexes

None

top

IDA_POLICY_VIOLATION_PREVIEWS Table

Structure

	IDA_POLICY_VIOLATION_PREVIEWS
PK	ID
	IDA_POLICY_VIOLATION_ID IDA_POLICY_ID USER_ID SEVERITY POLICY_VIOLATION_STATE CREATEDATE CREATE_SCAN_ID

Index

Index	PK	Unique	Keys
IDA_POLICY_VIOLATION_PREVIEWS	✓	Yes	ID

Description

The IDA_POLICY_VIOLATION_PREVIEWS table saves violation information that the user can preview without running the actual scan. Each entry has the following fields: IDA_POLICY_VIOLATION_ID, IDA_POLICY_ID, USER_ID, POLICY_VIOLATION_STATE, SEVERITY (1-High, 2-Medium, 3-Low), CREATEDATE, and CREATE_SCAN_ID. The POLICY_VIOLATION_STATE field stores the state of the violation in the form of an integer that can have the following values: 1-Open, 2-Closed, 3-Remediated, or 4-Mitigated.

Primary Keys

PK_IDA_VIOLATION_PREVIEWS - primary key on column ID

Foreign Keys

None

Indexes

None

top

IDA_VIOLATION_PREVIEW_CAUSES Table

Structure

	IDA_VIOLATION_PREVIEW_CAUSES
PK	ID
	IDA_POLICY_VIOLATION_ID IDA_RULE_ID IDA_RULE_VERSION_ID RBACX_OBJECT_ID RBACX_OBJECT_TYPE DESCRIPTION STATUS

Index

Index	PK	Unique	Keys
PK_IDA_VIOLATION_PREVIEW_CAUSES	✓	Yes	ID

Description

The IDA_VIOLATION_PREVIEW_CAUSES table saves information about the causes behind policy violations as detected during identity audit policy *previews*. Each entry has a unique ID. The IDA_POLICY_VIOLATION_ID field is derived from the IDA_POLICY_VIOLATION_PREVIEW table. The IDA_RULE_ID and IDA_RULE_VERSION_ID fields contain information about the rule set that the policy was run against, whereas the RBACX_OBJECT_ID and RBACX_OBJECT_TYPE fields give information about the Role Manager object against which the policy violation was detected. Additional information, such as the DESCRIPTION of the policy violation cause and the STATUS of the policy violation, is also stored in this table.

Primary Keys

1.PK_IDA_VIOLATION_PREVIEW_CAUSES - primary key on column ID

Foreign Keys

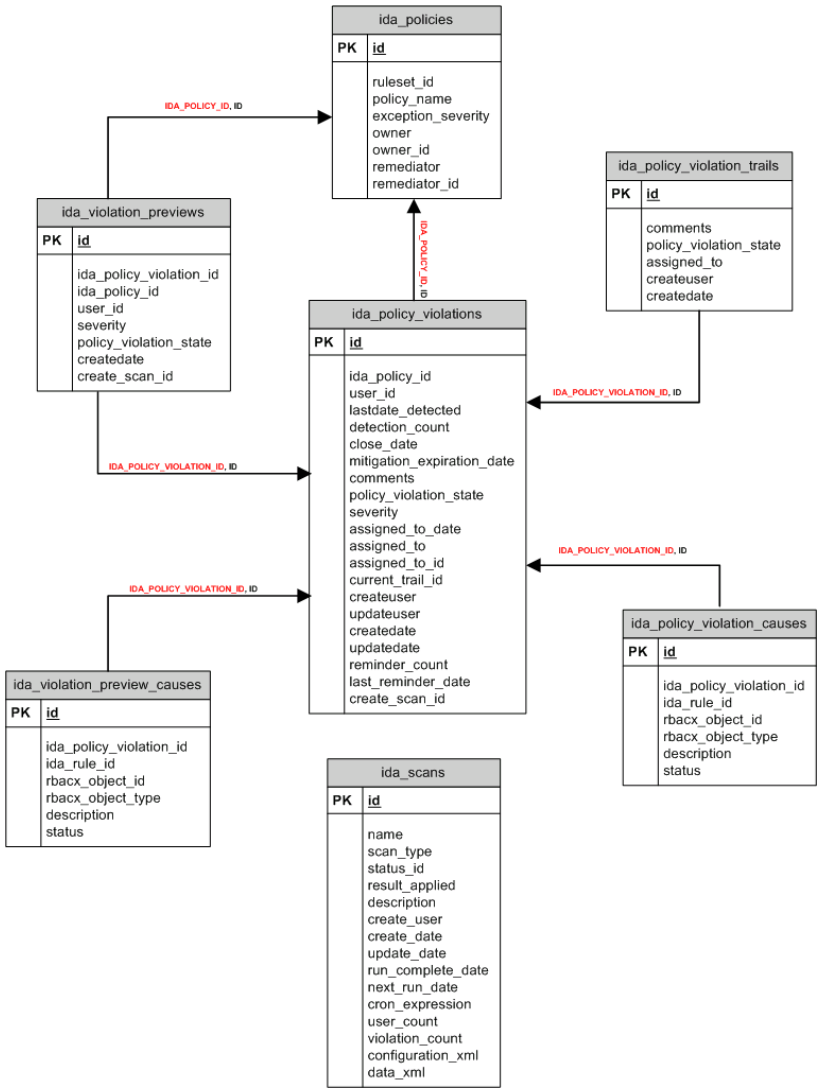
None

Indexes

None

top

Identity Audit Module Entity Relationship



top

Rule Engine Module

Tables in the Rule Engine Module

This chapter describes the tables that make up the Rule Engine module.

RULES Table

Structure

	RULES
PK	RULE_ID
I1	RULENAME DESCRIPTION STATUS_ID TYPE_ID CURRENT_VERSION_ID VERSION_COUNT RULE_CONDITION CREATEUSER UPDATEUSER CREATEDATE UPDATEDATE

Index

Index	PK	Unique	Keys
PK_RULES	✓	Yes	RULE_ID

Description

Role Manager can create rules that, based on attributes, assign users and roles to different components in the warehouse. For example, Role Manager rules can assign users and roles to business units based on HR attributes. The RULES table stores information about the rules that are created in Role Manager. Each rule is identified by a unique RULE_ID. The RULE_CONDITION field saves the condition that is used to define the rule. The STATUS_ID and CURRENT_VERSION_ID columns save the status and the version of the rule respectively. The TYPE_ID column identifies whether the rule is a role membership rule or a user business structure rule. In addition, the VERSION_COUNT, CREATEUSER, UPDATEUSER, CREATEDATE, and UPDATEDATE fields are used for audit purposes.

Primary Keys

1. PK_RULES - column RULE_ID

Foreign Keys

None

Indexes

None

top

RULE_OWNERS Table

Structure

	RULE_OWNERS
PK	ID
	RULE_ID RULE_VERSION_ID OWNER_ID OWNER_TYPE

Indexes

Index	PK	Unique	Keys
PK_RULE_OWNERS	✓	Yes	ID

Description

The RULE_OWNERS table defines the relationship between a rule and its owner (which can be either a Globaluser or a role in Role Manager). The RULE_ID field refers to the rule in the RULES table, whereas the OWNER_ID field refers to either the GLOBALUSERS table or the ROLES table. The OWNER_TYPE field is used to define if the owner is a global user or a role in Role Manager.

Primary Keys

PK_RULE_OWNERS on column ID

Foreign Keys

None

Indexes

None

top

RULE_SETS Table

Structure

	RULE_SETS
PK	ID
	STATUS_ID TYPE_ID RULESETNAME DESCRIPTION CREATEUSER UPDATEUSER CREATEDATE UPDATEDATE

Indexes

Index	PK	Unique	Keys
PK_RULE_SETS	✓	Yes	ID

Description

The RULE_SETS table contains grouping of rules. Every rule_set is identified by a unique ID. The RULESETNAME and DESCRIPTION fields give additional information about the rule set.

Primary Keys

PK_RULE_SETS - primary key on column ID

Foreign Keys

None

Indexes

None

top

RULE_SET_RULES Table

Structure

	RULE_SETS
	RULE_ID RULE_SET_ID

Indexes

None

Description

The RULE_SET_RULES table is a composite table that contains rule-sets-to-rule associations.

Primary Keys

None

Foreign Keys

None

Indexes

None

top

RULE_TYPES Table

Structure

	ARC_ACCOUNT_ATTRIBUTES
PK	ID
	LABEL DESCRIPTION

Index

Index	PK	Unique	Keys
PK_RULE_TYPES	✓	YES	ID

Description

The RULE_TYPES table is used to define a rule category. Two rule types are available:

- Role membership rule
- User BusinessStructure rule

Primary Keys

PK_RULE_TYPES on column ID

Foreign Keys

None

Indexes

None

top

RULE_VERSIONS Table

Structure

	RULE_VERSIONS
PK	VERSION_ID
I3	VERSION_STATUS_ID
I2	VERSION_NUMBER
I1	BASE_VERSION_ID
	WORKFLOW_ID
	RULE_ID
	RULENAME
	DESCRIPTION
	STATUS_ID
	TYPE_ID
	RULE_CONDITION
	CREATEUSER
	UPDATEUSER
	CREATEDATE
	UPDATEDATE
	LAST_APPROVER
	LAST_APPROVAL_DATE
	LAST_APPROVAL_COMMENTS

Index

Index	PK	Unique	Keys
PK_RULE_VERSIONS	✓	Yes	VERSION_ID

Description

When a condition is added or removed from a rule in Role Manager, a rule approval process is initiated by the system. A snapshot of the existing rule is captured and saved into the RULE_VERSIONS table prior to initiation of the rule approval process. This facilitates in maintaining a rule's history, which can be accessed to compare previous versions.

Primary Keys

PK_RULE_VERSIONS - primary key on column VERSION_ID

Foreign Keys

None

Indexes

None

top

ROLE_MEMBERSHIP_RULE Table

Structure

	ROLE_MEMBERSHIP_RULE

	ID
	NAME
	RULE_XML

Index

Index	PK	Unique	Keys
	✓		

Description

The ROLE_MEMBERSHIP_RULE table contains information about the rules generated to create associations between global users and roles. The RULE_XML field is a binary representation of the rule details.

Primary Keys

None

Foreign Keys

None

Indexes

None

top

ROLE_MEMBERSHIP_RULE_DATA Table

Structure

	ROLE_MEMBERSHIP_RULE_DATA
	RULE_ID
	RULE_VERSION_ID
	ROLE_ID
	REMOVAL_ACTION
	GRACE_PERIOD
	NOTIFICATION_TEMPLATE_ID

Index

None

Description

The ROLE_MEMBERSHIP_RULE_DATA stores information about which role to assign for a role provisioning rule.

Primary Keys

None

Foreign Keys

None

Indexes

None

top

ROLE_MEMBERSHIP_RULE_SCANS Table

Structure

	ROLE_MEMBERSHIP_RULE_SCANS
PK	**SCAN_ID**
	RULE_ID
	RULE_VERSION_ID
	STATUS_ID
	TYPE_ID
	RESULT_APPLIED
	JOB_NAME
	RULE_VERSION_PREVIEW
	ARCHIVAL_STATUS_ID
	ARCHIVAL_DATE
	DESCRIPTION
	CREATE_USER
	CREATE_DATE
	UPDATE_USER
	UPDATE_DATE
	DATA_XML

Index

Index	PK	Unique	Keys
PK_ROLE_MEMBERSHIP_RULE_SCANS	✓	YES	SCAN_ID

Description

The ROLE_MEMBERSHIP_RULE_SCANS tables saves information regarding the user set to be processed.

Primary Keys

PK_ROLE_MEMBERSHIP_RULE_SCANS - on column SCAN_ID

Foreign Keys

None

Indexes

None

top

RM_RULE_SCAN_RESULTS Table

Structure

	RM_RULE_SCAN_RESULTS
	RULE_SCAN_ID USERS_PROCESSED CREATE_DATE UPDATE_DATE DATA_XML

Index

None

Description

The RM_RULE_SCAN_RESULTS table stores information about the users to add or remove when a rule scan is processing users.

Primary Keys

None

Foreign Keys

None

Indexes

None

top

USER_BU_RULE_SCANS Table

Structure

	USER_BU_RULE_SCANS
PK	ID
	STATUS_ID JOB_NAME USER_SELECTION_CRITERIA_XML BUSINESSUNIT_IDS_XML USER_IDS_XML USER_FILTER_XML ARCHIVAL_STATUS_ID ARCHIVAL_DATE RESULT_APPLIED DESCRIPTION RULE_ID RULE_VERSION_ID CREATE_USER CREATE_DATE UPDATE_USER UPDATE_DATE SCAN_RESULT_ID TYPE_ID TOTAL_USERS

Index

Index	PK	Unique	Keys
PK_USER_BU_RULE_SCANS	✓	YES	ID

Description

The USER_BU_RULE_SCANS table stores information about the user set to be processed, the rule to used during the scan, and the business unit that is associated with the user. In addition, it contains general information about the scan, such as a summary and the state of the scan.

Primary Keys

PK_USER_BU_RULE_SCANS - on column ID

Foreign Keys

None

Indexes

None

top

USER_BU_RULE_SCAN_RESULTS Table

Structure

	USER_BU_RULE_SCAN_RESULTS
PK	ID
	RULE_SCAN_ID CREATE_DATE ASSIGNED_USERS_MAP_XML UN_ASSIGNED_USERS_MAP_XML EXISTING_ASSIGNED_MAP_XML USERS_PROCESSED

Index

None

Description

The USER_BU_RULE_SCAN_RESULTS table stores information about the results of completed scans as defined in the USER_BU_RULE_SCAN_RESULTS table. This table stores information about users that are assigned and unassigned to business units as a result of the scan.

Primary Keys

PK_USER_BU_RULE_SCAN_RESULTS - on column ID

Foreign Keys

None

Indexes

None

top

USER_BU_RULE_EXTENDED_ATTS Table

Structure

	USER_BU_RULE_EXTENDED_ATTS
PK	ID
	REMOVAL_ACTION BUSINESS_UNIT_ID GRACE_PERIOD NOTIFICATION_TEMPLATE_ID

Index

Index	PK	Unique	Keys
PK_USER_BU_RULE_EXTENDED_ATTS	✓	YES	ID

Description

The USER_BU_RULE_EXTENDED_ATTS table stores additional information about the business-structure-to-user association rules.

Primary Keys

PK_USER_BU_RULE_SCAN_RESULTS - on column ID

Foreign Keys

None

Indexes

None

top

USER_APPLICATION_SCANS Table

Structure

	USER_APPLICATION_SCANS
PK	ID
	START_DATE END_DATE SUCCESS SCAN_TYPE MEMBERS_FOUND MEMBERS_REMOVED

Index

Index	PK	Unique	Keys
PK_AMES	✓	YES	ID

Description

The USER_APPLICATION_SCANS table stores information about application scans, such as the scan's start date and end date, members found and removed, and summary information.

Primary Keys

PK_AMES - on column ID

Foreign Keys

None

Indexes

None

top

USER_TRANSFERS Table

Structure

	USER_TRANSFERS
PK	ID
	GLOBALUSER_ID SRM_TRANSFER_DATE

Index

None

Description

The USER_TRANSFERS table holds users for which there is a change in some predefined property. Information stored in this table includes the unique ID of the globaluser and the transfer date.

Primary Keys

None

Foreign Keys

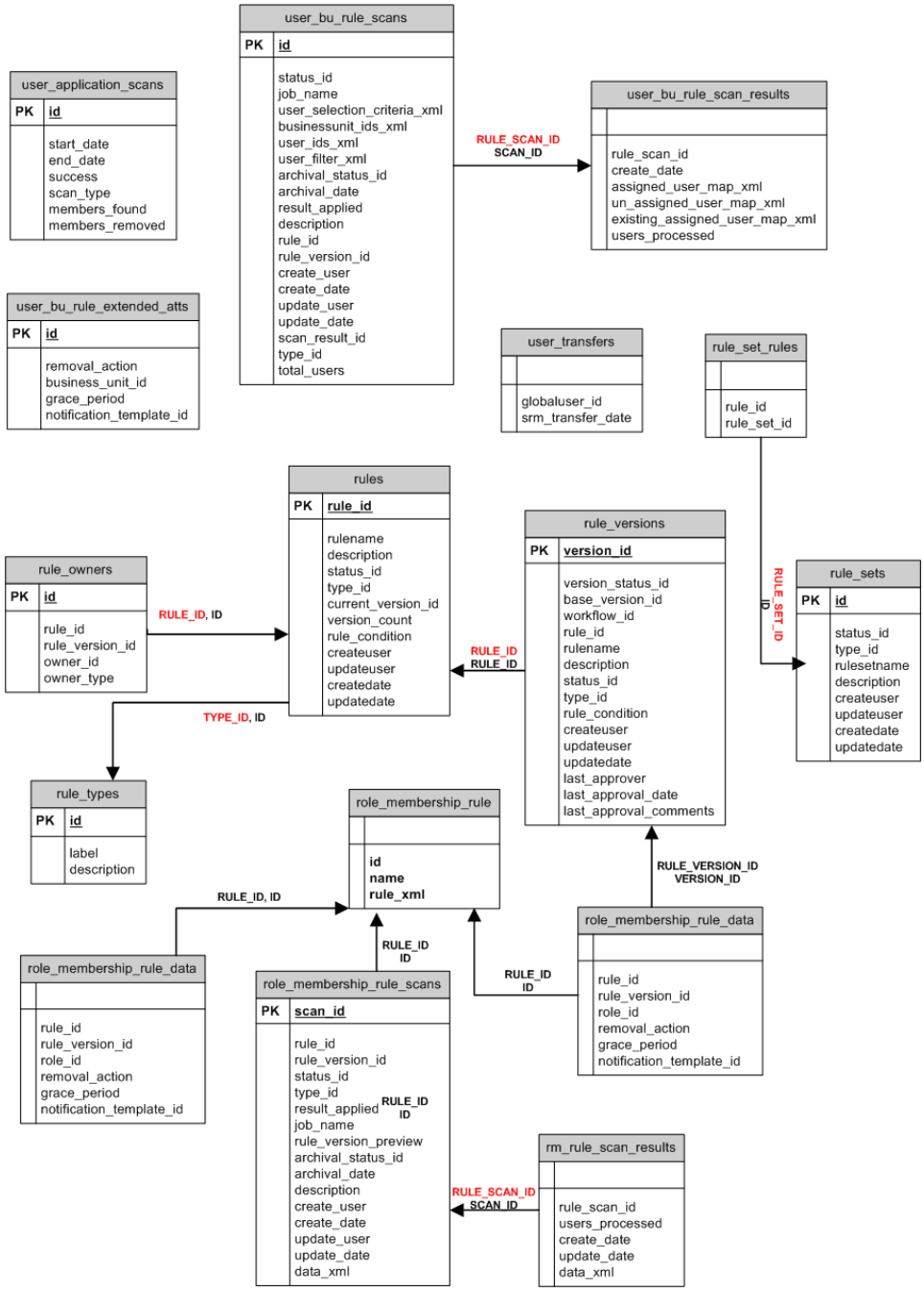
None

Indexes

None

top

Rule Engine Module Entity Relationship



top

Workflow Module

Tables in the Workflow Module

This chapter describes the tables that make up the Role Manager Workflow module.

OS_WFENTRY Table

Structure

	OS_WFENTRY
PK	ID
	NAME STATE CALLER START_DATE

Indexes

Index	PK	Unique	Keys
PK_WFENTRY	✔		ID

Description

The OS_WFENTRY table stores workflow instances, along with the NAME, STATE, CALLER, and START_DATE information for each instance.

Primary Keys

1.ID - primary key on column ID

Foreign Keys

None

Indexes

None

top

OS_CURRENTSTEP Table

Structure

	OS_CURRENTSTEP
PK	ID
FK	ENTRY_ID STEP_ID ACTION_ID OWNER START_DATE FINISH_DATE DUE_DATE STATUS CALLER

Indexes

Index	PK	Unique	Keys
PK_CRTSP	✓		ID
FK_CRTSP_WFNTR_ID			ENTRY_ID

Description

The OS_CURRENTSTEP table holds all of the steps for a workflow that is in progress. This table references the OS_WFENTRY table's ID column using ENTRY_ID as foreign key to establish the association.

Primary Keys

1.PK_CRTSP - primary key on column ID.

Foreign Keys

1.FK_CRTSP_WFNTR_ID - foreign key on column ENTRY_ID refers to ID column of OS_WFENTRY table.

Indexes

None

top

OS_HISTORYSTEP Table

Structure

	OS_HISTORYSTEP
PK	ID
FK1	ENTRY_ID STEP_ID ACTION_ID OWNER START_DATE FINISH_DATE DUE_DATE STATUS CALLER

Indexes

Index	PK	Unique	Keys
PK_HTRSP	✓		ID
FK_HSTSTP_FENTR_ID			ENTRY_ID

Description

The OS_HISTORYSTEP table saves the historical steps taken duing a workflow instance. This table references the OS_WFENTRY table's ID column using ENTRY_ID as the foreign key to establish the association.

Primary Keys

1.PK_HTRSP - primary key on column ID.

Foreign Keys

1.FK_HSTSTP_FENTR_ID - foreign key on column ENTRY_ID refers to ID column of OS_WFENTRY table.

Indexes

None

top

OS_CURRENTSTEP_PREV Table

Structure

	OS_CURRENTSTEP_PREV
PK,FK1	ID
PK,FK2	PREVIOUS_ID

Indexes

Index	PK	Unique	Keys
PK_CRTSP_PRV	✓		ID, PREVIOUS_ID
FK_CRTSP_PRV_HTRSP_ID			PREVIOUS_ID

Description

The OS_CURRENTSTEP_PREV table saves the previous steps taken for a workflow instance that is in progress. It refers to both the OS_CURRENTSTEP and the OS_HISTORYSTEP tables via foreign keys.

Primary Keys

1. PK_CRTSP_PRV - composite primary key on column ID, PREVIOUS_ID

Foreign Keys

1.FK_CRSTP_PRV_CRTSP_ID - foreign key on column ID refers to ID column of OS_CURRENTSTEP table.
2.FK_CRSTP_PRV_HTRSP_ID - foreign key on column PREVIOUS_ID refers to ID column of OS_HISTORYSTEP table.

Indexes

None

top

OS_HISTORYSTEP_PREV Table

Structure

	OS_HISTORYSTEP_PREV
PK,FK1	ID
PK,FK2	PREVIOUS_ID

Indexes

Index	PK	Unique	Keys
PK_HTRSP_PREV	✓		ID, PREVIOUS_ID
FK_HTRSP_PREV_ID_HTRSP_ID			PREVIOUS_ID

Description

The OS_HISTORYSTEP_PREV table saves the previous steps trail that is recorded in the OS_HISTORYSTEPS table. It refers to the OS_HISTORYSTEP tables via foreign keys.

Primary Keys

1.PK_HTRSP_PREV - composite primary key on column ID, PREVIOUS_ID.

Foreign Keys

1.FK_HTRSP_PREV_HTRSP_ID - foreign key on column ID refers to ID column of OS_HISTORYSTEP table.
2.FK_HTRSP_PREV_ID_HTRSP_ID - foreign key on column PREVIOUS_ID refers to ID column of OS_HISTORYSTEP table.

Indexes

None

top

OS_PROPERTYENTRY Table

Structure

	OS_PROPERTYENTRY
PK	ITEM_KEY
PK	ENTITY_ID
	ENTITY_NAME ITEM_TYPE STRING_VALUE DATE_VALUE DATA_VALUE FLOAT_VALUE NUMBER_VALUE INTEGER_VALUE LONG_VALUE BOOLEAN_VALUE OBJECT_VALUE

Indexes

Index	PK	Unique	Keys
PK_OS_PROPERTYENTRY	✓		ITEM_KEY, ENTITY_ID

Description

The Role Manager workflow engine stores values such as role ID and role version ID in the OS_PROPERTYENTRY table.

Primary Keys

1.PK_OS_PROPERTYENTRY - Composite primary key on column ITEM_KEY, ENTITY_ID.

Foreign Keys

None

Indexes

None

top

APPROVAL_REQUESTS Table

Structure

	APPROVAL_REQUESTS
PK	ID
	REQUESTED_BY REQUEST_DATE REQUEST_STATUS REQUEST_TYPE WORKFLOW_ID OBJECT_ID VERSION_ID VERSION_NUMBER OBJECT_NAME OBJECT_TYPE

Indexes

Index	PK	Unique	Keys
PK_APPROVAL_REQUESTS	✓		ID

Description

The APPROVAL_REQUESTS table holds information used during a role approval process. An approval process is initiated when a policy is either added to a role or removed from a role, or a user changes their role membership. An approval process is associated by a role versioning process that maintains a trail of modifications made to a role. The columns VERSION_ID and VERSION_NUMBER refer to the versioning of roles.

Primary Keys

1.PK_APPROVAL_REQUESTS - composite primary key on columns ID.

Foreign Keys

None

Indexes

None

top

APPROVAL_REQUEST_HISTORY Table

Structure

	APPROVAL_REQUEST_HISTORY
I1	REQUEST_ID STEP_ID OBJECT_ID OBJECT_NAME OBJECT_TYPE ACTION ACTION_DATE APPROVER APPROVER_COMMENTS

Indexes

Index	PK	Unique	Keys
IX_APPROVAL_REQUEST_HISTORY		No	REQUEST_ID

Description

The APPROVAL_REQUEST_HISTORY table maintains the history of actions taken during each step of the role versioning and membership approval processes.

Primary Keys

None

Foreign Keys

None

Indexes

1.IX_APPROVAL_REQUEST_HISTORY - non-unique index on column REQUEST_ID.

top

ROLE_MATCHING_ANALYSIS Table

Structure

	ROLE_MATCHING_ANALYSIS
PK	APPROVAL_REQUEST_ID

	RESULTXML

Indexes

Index	PK	Unique	Keys
PK_ROLE_MATCHING_ANALYSIS	✓		APPROVAL_REQUEST_ID

Description

When a role is modified in Role Manager, either because a policy was added or removed, or because there was a change in role membership, the modified role is compared with the existing roles in the system. The APPROVAL_REQUEST_ID column associates the ROLE_MATCHING_ANALYSIS table with the APPROVAL_REQUESTS table while the analysis result is saved in the resultXml column in XML format.

Primary Keys

1.PK_ROLE_MATCHING_ANALYSIS - primary key on columns APPROVAL_REQUEST_ID.

Foreign Keys

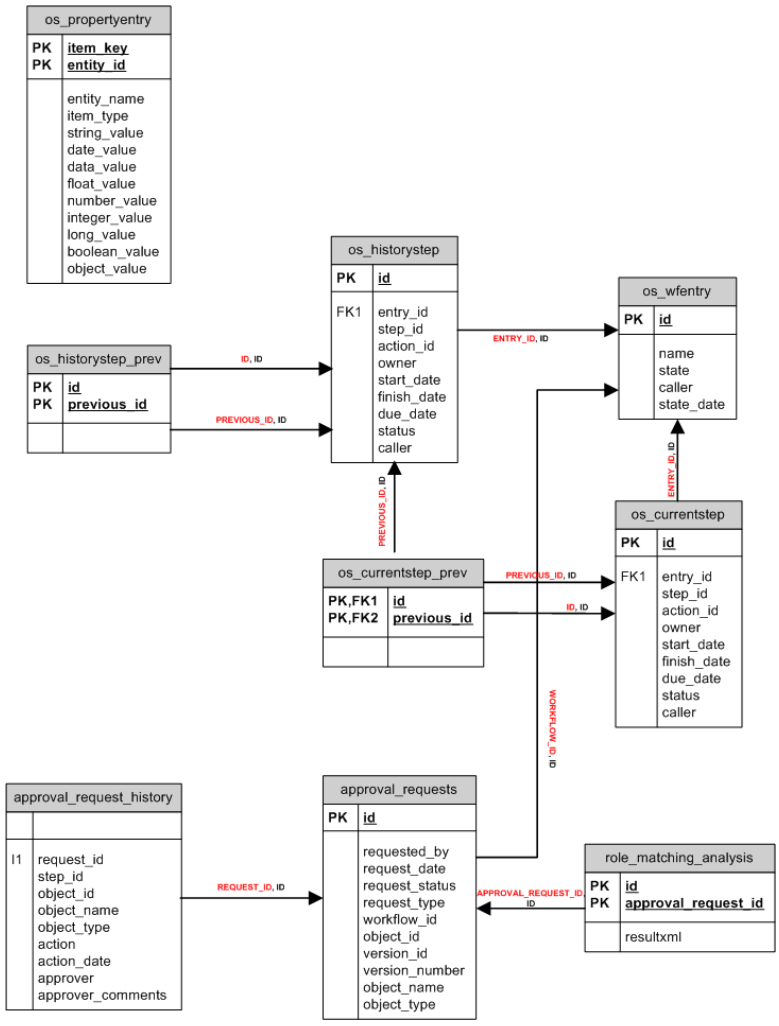
None

Indexes

None

top

Workflow Module Entity Relationship



top

Scheduling Module

Tables in the Workflow Module

This chapter describes the tables that make up the Role Manager Workflow module.

OS_WFENTRY Table

Structure

	OS_WFENTRY
PK	ID
	NAME STATE CALLER

START_DATE

Indexes

Index	PK	Unique	Keys
PK_WFENTRY	✔		ID

Description

The OS_WFENTRY table stores workflow instances, along with the NAME, STATE, CALLER, and START_DATE information for each instance.

Primary Keys

1.ID - primary key on column ID

Foreign Keys

None

Indexes

None

top

OS_CURRENTSTEP Table

Structure

	OS_CURRENTSTEP
PK	**ID**
FK	**ENTRY_ID** STEP_ID ACTION_ID OWNER START_DATE FINISH_DATE DUE_DATE STATUS CALLER

Indexes

Index	PK	Unique	Keys
PK_CRTSP	✓		ID
FK_CRTSP_WFNTR_ID			ENTRY_ID

Description

The OS_CURRENTSTEP table holds all of the steps for a workflow that is in progress. This table references the OS_WFENTRY table's ID column using ENTRY_ID as foreign key to establish the association.

Primary Keys

1.PK_CRTSP - primary key on column ID.

Foreign Keys

1.FK_CRTSP_WFNTR_ID - foreign key on column ENTRY_ID refers to ID column of OS_WFENTRY table.

Indexes

None

top

OS_HISTORYSTEP Table

Structure

	OS_HISTORYSTEP
PK	**ID**
FK1	**ENTRY_ID** STEP_ID ACTION_ID OWNER START_DATE FINISH_DATE DUE_DATE STATUS CALLER

Indexes

Index	PK	Unique	Keys
PK_HTRSP	✓		ID
FK_HSTSTP_FENTR_ID			ENTRY_ID

Description

The OS_HISTORYSTEP table saves the historical steps taken duing a workflow instance. This table references the OS_WFENTRY table's ID column using ENTRY_ID as the foreign key to establish the association.

Primary Keys

1.PK_HTRSP - primary key on column ID.

Foreign Keys

1.FK_HSTSTP_FENTR_ID - foreign key on column ENTRY_ID refers to ID column of OS_WFENTRY table.

Indexes

None

top

OS_CURRENTSTEP_PREV Table

Structure

	OS_CURRENTSTEP_PREV
PK,FK1	**ID**
PK,FK2	**PREVIOUS_ID**

Indexes

Index	PK	Unique	Keys
PK_CRTSP_PRV	✓		ID, PREVIOUS_ID
FK_CRTSP_PRV_HTRSP_ID			PREVIOUS_ID

Description

The OS_CURRENTSTEP_PREV table saves the previous steps taken for a workflow instance that is in progress. It refers to both the OS_CURRENTSTEP and the OS_HISTORYSTEP tables via foreign keys.

Primary Keys

1. PK_CRTSP_PRV - composite primary key on column ID, PREVIOUS_ID

Foreign Keys

1.FK_CRSTP_PRV_CRTSP_ID - foreign key on column ID refers to ID column of OS_CURRENTSTEP table.
2.FK_CRSTP_PRV_HTRSP_ID - foreign key on column PREVIOUS_ID refers to ID column of OS_HISTORYSTEP table.

Indexes

None

top

OS_HISTORYSTEP_PREV Table

Structure

	OS_HISTORYSTEP_PREV
PK,FK1	**ID**
PK,FK2	**PREVIOUS_ID**

Indexes

Index	PK	Unique	Keys
PK_HTRSP_PREV	✓		ID, PREVIOUS_ID
FK_HTRSP_PREV_ID_HTRSP_ID			PREVIOUS_ID

Description

The OS_HISTORYSTEP_PREV table saves the previous steps trail that is recorded in the OS_HISTORYSTEPS table. It refers to the OS_HISTORYSTEP tables via foreign keys.

Primary Keys

1.PK_HTRSP_PREV - composite primary key on column ID, PREVIOUS_ID.

Foreign Keys

1.FK_HTRSP_PREV_HTRSP_ID - foreign key on column ID refers to ID column of OS_HISTORYSTEP table.
2.FK_HTRSP_PREV_ID_HTRSP_ID - foreign key on column PREVIOUS_ID refers to ID column of OS_HISTORYSTEP table.

Indexes

None

top

OS_PROPERTYENTRY Table

Structure

	OS_PROPERTYENTRY
PK	**ITEM_KEY**
PK	**ENTITY_ID**
	ENTITY_NAME ITEM_TYPE STRING_VALUE DATE_VALUE DATA_VALUE FLOAT_VALUE NUMBER_VALUE INTEGER_VALUE LONG_VALUE BOOLEAN_VALUE OBJECT_VALUE

Indexes

Index	PK	Unique	Keys
PK_OS_PROPERTYENTRY	✓		ITEM_KEY, ENTITY_ID

Description

The Role Manager workflow engine stores values such as role ID and role version ID in the OS_PROPERTYENTRY table.

Primary Keys

1.PK_OS_PROPERTYENTRY - Composite primary key on column ITEM_KEY, ENTITY_ID.

Foreign Keys

None

Indexes

None

top

APPROVAL_REQUESTS Table

Structure

	APPROVAL_REQUESTS
PK	ID
	REQUESTED_BY REQUEST_DATE REQUEST_STATUS REQUEST_TYPE WORKFLOW_ID OBJECT_ID VERSION_ID VERSION_NUMBER OBJECT_NAME OBJECT_TYPE

Indexes

Index	PK	Unique	Keys
PK_APPROVAL_REQUESTS	✓		ID

Description

The APPROVAL_REQUESTS table holds information used during a role approval process. An approval process is initiated when a policy is either added to a role or removed from a role, or a user changes their role membership. An approval process is associated by a role versioning process that maintains a trail of modifications made to a role. The columns VERSION_ID and VERSION_NUMBER refer to the versioning of roles.

Primary Keys

1.PK_APPROVAL_REQUESTS - composite primary key on columns ID.

Foreign Keys

None

Indexes

None

top

APPROVAL_REQUEST_HISTORY Table

Structure

	APPROVAL_REQUEST_HISTORY
I1	REQUEST_ID STEP_ID OBJECT_ID OBJECT_NAME OBJECT_TYPE ACTION ACTION_DATE APPROVER APPROVER_COMMENTS

Indexes

Index	PK	Unique	Keys
IX_APPROVAL_REQUEST_HISTORY		No	REQUEST_ID

Description

The APPROVAL_REQUEST_HISTORY table maintains the history of actions taken during each step of the role versioning and membership approval processes.

Primary Keys

None

Foreign Keys

None

Indexes

1.IX_APPROVAL_REQUEST_HISTORY - non-unique index on column REQUEST_ID.

top

ROLE_MATCHING_ANALYSIS Table

Structure

	ROLE_MATCHING_ANALYSIS
PK	APPROVAL_REQUEST_ID
	RESULTXML

Indexes

Index	PK	Unique	Keys
PK_ROLE_MATCHING_ANALYSIS	✓		APPROVAL_REQUEST_ID

Description

When a role is modified in Role Manager, either because a policy was added or removed, or because there was a change in role membership, the modified role is compared with the existing roles in the system. The APPROVAL_REQUEST_ID column associates the ROLE_MATCHING_ANALYSIS table with the APPROVAL_REQUESTS table while the analysis result is saved in the resultXml column in XML format.

Primary Keys

1.PK_ROLE_MATCHING_ANALYSIS - primary key on columns APPROVAL_REQUEST_ID.

Foreign Keys

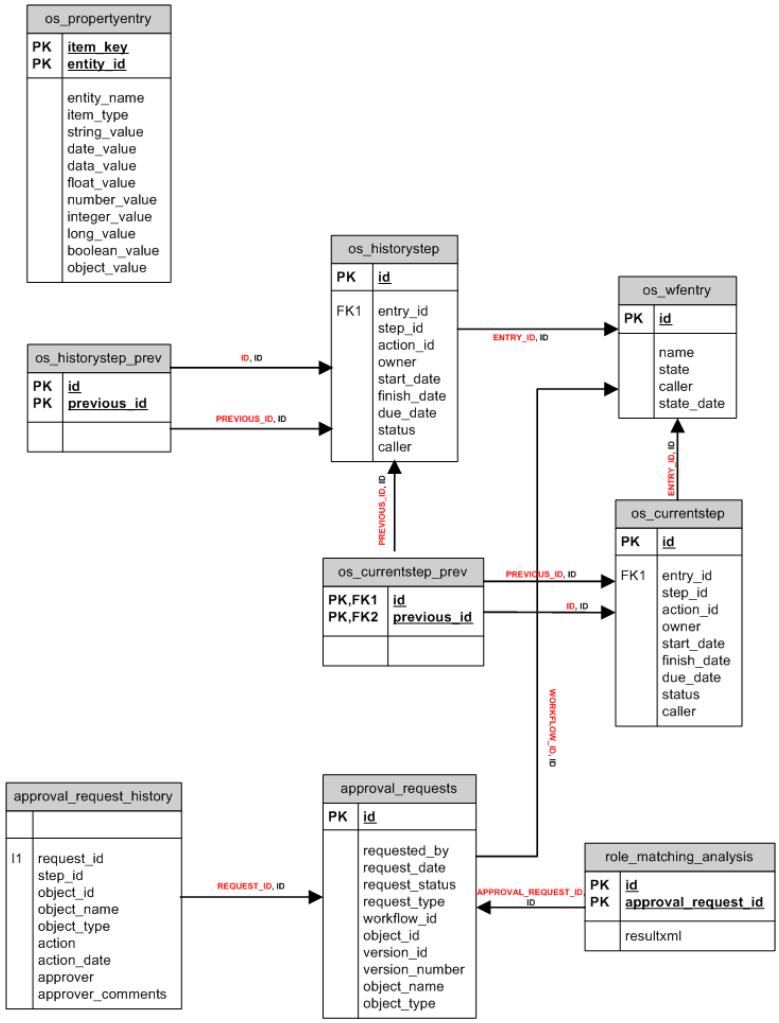
None

Indexes

None

top

Workflow Module Entity Relationship



top

Reporting Module

Tables in the Workflow Module

This chapter describes the tables that make up the Role Manager Workflow module.

OS_WFENTRY Table

Structure

	OS_WFENTRY
PK	ID
	NAME
	STATE
	CALLER
	START_DATE

Indexes

Index	PK	Unique	Keys
PK_WFENTRY	✔		ID

Description

The OS_WFENTRY table stores workflow instances, along with the NAME, STATE, CALLER, and START_DATE information for each instance.

Primary Keys

1.ID - primary key on column ID

Foreign Keys

None

Indexes

None

top

OS_CURRENTSTEP Table

Structure

	OS_CURRENTSTEP
PK	ID
FK	ENTRY_ID STEP_ID ACTION_ID OWNER START_DATE FINISH_DATE DUE_DATE STATUS CALLER

Indexes

Index	PK	Unique	Keys
PK_CRTSP	✓		ID
FK_CRTSP_WFNTR_ID			ENTRY_ID

Description

The OS_CURRENTSTEP table holds all of the steps for a workflow that is in progress. This table references the OS_WFENTRY table's ID column using ENTRY_ID as foreign key to establish the association.

Primary Keys

1.PK_CRTSP - primary key on column ID.

Foreign Keys

1.FK_CRTSP_WFNTR_ID - foreign key on column ENTRY_ID refers to ID column of OS_WFENTRY table.

Indexes

None

top

OS_HISTORYSTEP Table

Structure

	OS_HISTORYSTEP
PK	ID
FK1	ENTRY_ID STEP_ID ACTION_ID OWNER START_DATE FINISH_DATE DUE_DATE STATUS CALLER

Indexes

Index	PK	Unique	Keys
PK_HTRSP	✓		ID
FK_HSTSTP_FENTR_ID			ENTRY_ID

Description

The OS_HISTORYSTEP table saves the historical steps taken duing a workflow instance. This table references the OS_WFENTRY table's ID column using ENTRY_ID as the foreign key to establish the association.

Primary Keys

1.PK_HTRSP - primary key on column ID.

Foreign Keys

1.FK_HSTSTP_FENTR_ID - foreign key on column ENTRY_ID refers to ID column of OS_WFENTRY table.

Indexes

None

top

OS_CURRENTSTEP_PREV Table

Structure

	OS_CURRENTSTEP_PREV

PK,FK1	ID
PK,FK2	PREVIOUS_ID

Indexes

Index	PK	Unique	Keys
PK_CRTSP_PRV	✓		ID, PREVIOUS_ID
FK_CRTSP_PRV_HTRSP_ID			PREVIOUS_ID

Description

The OS_CURRENTSTEP_PREV table saves the previous steps taken for a workflow instance that is in progress. It refers to both the OS_CURRENTSTEP and the OS_HISTORYSTEP tables via foreign keys.

Primary Keys

1. PK_CRTSP_PRV - composite primary key on column ID, PREVIOUS_ID

Foreign Keys

1.FK_CRSTP_PRV_CRTSP_ID - foreign key on column ID refers to ID column of OS_CURRENTSTEP table.
2.FK_CRSTP_PRV_HTRSP_ID - foreign key on column PREVIOUS_ID refers to ID column of OS_HISTORYSTEP table.

Indexes

None

top

OS_HISTORYSTEP_PREV Table

Structure

	OS_HISTORYSTEP_PREV
PK,FK1	ID
PK,FK2	PREVIOUS_ID

Indexes

Index	PK	Unique	Keys
PK_HTRSP_PREV	✓		ID, PREVIOUS_ID
FK_HTRSP_PREV_ID_HTRSP_ID			PREVIOUS_ID

Description

The OS_HISTORYSTEP_PREV table saves the previous steps trail that is recorded in the OS_HISTORYSTEPS table. It refers to the OS_HISTORYSTEP tables via foreign keys.

Primary Keys

1.PK_HTRSP_PREV - composite primary key on column ID, PREVIOUS_ID.

Foreign Keys

1.FK_HTRSP_PREV_HTRSP_ID - foreign key on column ID refers to ID column of OS_HISTORYSTEP table.
2.FK_HTRSP_PREV_ID_HTRSP_ID - foreign key on column PREVIOUS_ID refers to ID column of OS_HISTORYSTEP table.

Indexes

None

top

OS_PROPERTYENTRY Table

Structure

	OS_PROPERTYENTRY
PK	ITEM_KEY
PK	ENTITY_ID
	ENTITY_NAME ITEM_TYPE STRING_VALUE DATE_VALUE DATA_VALUE FLOAT_VALUE NUMBER_VALUE INTEGER_VALUE LONG_VALUE BOOLEAN_VALUE OBJECT_VALUE

Indexes

Index	PK	Unique	Keys
PK_OS_PROPERTYENTRY	✓		ITEM_KEY, ENTITY_ID

Description

The Role Manager workflow engine stores values such as role ID and role version ID in the OS_PROPERTYENTRY table.

Primary Keys

1.PK_OS_PROPERTYENTRY - Composite primary key on column ITEM_KEY, ENTITY_ID.

Foreign Keys

None

Indexes

None

top

APPROVAL_REQUESTS Table

Structure

	APPROVAL_REQUESTS
PK	ID
	REQUESTED_BY REQUEST_DATE REQUEST_STATUS REQUEST_TYPE WORKFLOW_ID OBJECT_ID VERSION_ID VERSION_NUMBER OBJECT_NAME OBJECT_TYPE

Indexes

Index	PK	Unique	Keys
PK_APPROVAL_REQUESTS	✓		ID

Description

The APPROVAL_REQUESTS table holds information used during a role approval process. An approval process is initiated when a policy is either added to a role or removed from a role, or a user changes their role membership. An approval process is associated by a role versioning process that maintains a trail of modifications made to a role. The columns VERSION_ID and VERSION_NUMBER refer to the versioning of roles.

Primary Keys

1.PK_APPROVAL_REQUESTS - composite primary key on columns ID.

Foreign Keys

None

Indexes

None

top

APPROVAL_REQUEST_HISTORY Table

Structure

	APPROVAL_REQUEST_HISTORY
I1	REQUEST_ID STEP_ID OBJECT_ID OBJECT_NAME OBJECT_TYPE ACTION ACTION_DATE APPROVER APPROVER_COMMENTS

Indexes

Index	PK	Unique	Keys
IX_APPROVAL_REQUEST_HISTORY		No	REQUEST_ID

Description

The APPROVAL_REQUEST_HISTORY table maintains the history of actions taken during each step of the role versioning and membership approval processes.

Primary Keys

None

Foreign Keys

None

Indexes

1.IX_APPROVAL_REQUEST_HISTORY - non-unique index on column REQUEST_ID.

top

ROLE_MATCHING_ANALYSIS Table

Structure

	ROLE_MATCHING_ANALYSIS
PK	APPROVAL_REQUEST_ID
	RESULTXML

Indexes

Index	PK	Unique	Keys
PK_ROLE_MATCHING_ANALYSIS	✓		APPROVAL_REQUEST_ID

Description

When a role is modified in Role Manager, either because a policy was added or removed, or because there was a change in role membership, the modified role is compared with the existing roles in the system. The APPROVAL_REQUEST_ID column associates the ROLE_MATCHING_ANALYSIS table with the APPROVAL_REQUESTS table while the analysis result is saved in the resultXml column in XML format.

Primary Keys

1.PK_ROLE_MATCHING_ANALYSIS - primary key on columns APPROVAL_REQUEST_ID.

Foreign Keys

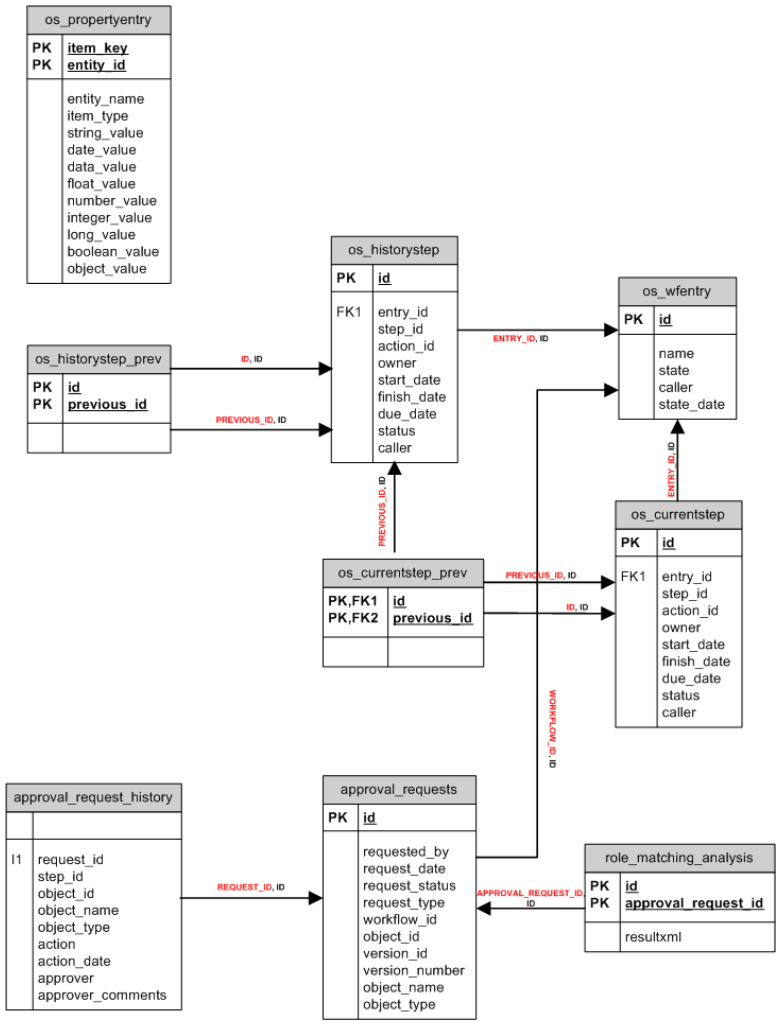
None

Indexes

None

top

Workflow Module Entity Relationship



top

Security Module

Tables in the Security Module

This chapter describes the tables that make up the Security module.

RBX_USERS Table

Structure

	RBX_USERS
PK	RBACXUSERKEY
	USERNAME PASSWORD ENABLED FIRSTNAME LASTNAME EMAIL LAST_PASSWORD_UPDATE USER_ID

Indexes

Index	PK	Unique	Keys
PK_RBACXUSER	✓	Yes	RBACXUSERKEY
IX_RBX_USERS		No	USER_ID

Description

A user who uses the Role Manager user interface may or may not be a part of a provisioning solution, therefore these users are defined in a different table, RBX_USERS. Each user is identified by unique entries in the RBACXUSERKEY and USERNAME columns, and an encrypted password in the PASSWORD column. The LAST_PASSWORD_UPDATE field saves the date that the password was last updated. In addition, details such as FIRSTNAME, LASTNAME, and EMAIL are

stored in this table, while the ENABLED field represents the status of the user.

The USER_ID refers to the globaluserkey of the corresponding globaluser created where the user is part of a provisioning system. One such example is the case of automated RBX_USER creation on certificate generation.

Primary Keys

1.PK_RBACXUSER - primary key on column RBACXUSERKEY

Foreign Keys

None

Indexes

1.IX_RBX_USERS - non-unique index on column USER_ID

top

PROXY_ASSIGNMENTS Table

Structure

	PROXY_ASSIGNMENTS
PK	ID
	ORIG_USER_ID PROXY_USER_ID START_DATE END_DATE NAME DESCRIPTION REQUEST_TYPE CREATEDATE UPDATEDATE CREATEUSER UPDATEUSER

Index

Index	PK	Unique	Keys
PK_PROXY_ASSIGNMENTS	✓	Yes	ID

Description

In Role Manager a designated proxy user can log in to the system and perform various operations on behalf of another user. This user association is based on the Global User ID field. The ORIG_USER_ID column contains the user ID of the original user, whereas the PROXY_USER_ID column contains the user ID of the assignee. Apart from defining the proxy user, the START_DATE, END_DATE, and REQUEST_TYPE columns in this table can be used to hold additional information for audit purposes.

Primary Keys

None

Foreign Keys

None

Indexes

None

top

PROXY_RBACXROLE Table

Structure

	PROXY_RBACXROLE
	PROXY_ID RBACXROLE_ID BUSINESSUNIT_ID

Index

Index	PK	Unique	Keys
PROXY_ID_RBACXROLE_IDX		Yes	PROXY_ID, RBACXROLE_ID, BUSINESSUNIT_ID
PROXY_ID_IDX		No	PROXY_ID
RBACXROLE_ID_IDX		No	RBACXROLE_ID
BUSINESSUNIT_ID_IDX		No	BUSINESSUNIT_ID

Description

The PROXY_RBACXROLE table saves information about the role that is granted to the proxy user as a result of the proxy assignment.

Primary Keys

None

Foreign Keys

None

Indexes

1. PROXY_ID_RBACXROLE_IDX - unique index on columns PROXY_ID, RBACXROLE_ID, BUSINESSUNIT_ID

2.PROXY_ID_IDX - non-unique index on column PROXY_ID

3.RBACXROLE_ID_IDX - non-unique index on column RBACXROLE_ID

4.BUSINESSUNIT_ID_IDX - non-unique index on column BUSINESSUNIT_ID

top

RBX_ROLES Table

Structure

	RBX_ROLES
PK	**RBACXROLEKEY**
	NAME
	DESCRIPTION
I1	SHORT_NAME
I2	PREDEFINED
	DELEGABLE

Indexes

Index	PK	Unique	Keys
PK_RBACXROLE	✓		RBACXROLEKEY
IX_RBX_ROLES_SHORT_NAME		Yes	SHORT_NAME
IX_RBX_ROLES_USER_PREDEF		No	PREDEFINED

Description

Roles are defined in the security module so that Role Manager can restrict access to the user interface based on access levels. These roles are stored in the RBX_ROLES table. Each role has a unique key defined in the RBACXROLEKEY column, and role details are stored in the NAME and DESCRIPTION fields. The SHORT_NAME and PREDEFINED fields are used in support of out-of-the-box RBX ROLES. The DELEGABLE field identifies whether the role can be delegated to another user.

Primary Keys

1.PK_RBACXROLE - primary key on column RBACXROLEKEY

Foreign Keys

None

Indexes

1.IX_RBX_ROLES_SHORT_NAME - unique index on column SHORT_NAME
2.IX_RBX_ROLES_USER_PREDEF - non-unique index on column PREDEFINED

top

PROXY_RBACXROLE Table

Structure

	PROXY_RBACXROLES
	PROXY_ID
	RBACXROLE_ID
	BUSINESSUNIT_ID

Indexes

Index	PK	Unique	Keys
PK_RBACXROLE	✓		RBACXROLEKEY
IX_RBX_ROLES_SHORT_NAME		Yes	SHORT_NAME
IX_RBX_ROLES_USER_PREDEF		No	PREDEFINED

Description

Primary Keys

1.PK_RBACXROLE - primary key on column RBACXROLEKEY

Foreign Keys

None

Indexes

1.IX_RBX_ROLES_SHORT_NAME - unique index on column SHORT_NAME
2.IX_RBX_ROLES_USER_PREDEF - non-unique index on column PREDEFINED

top

RBX_USER_RBX_ROLES Table

Structure

	RBX_USER_RBX_ROLES
PK	**RBACXUSERRBACXROLEKEY**
	RBACXUSERKEY
	RBACXROLEKEY

Indexes

Index	PK	Unique	Keys
PK_RBACXUSERRBACXROLES	✓	Yes	RBACXUSERRBACXROLEKEY

Description

A derived table, RBX_USER_RBX_ROLES, associates Role Manager users with their respective security roles. This table carries a unique key, RBACXUSERRBACXROLEKEY, and reference keys to the RBX_USERS and RBX_ROLES tables.

Primary Keys

1.PK_RBACXUSERRBACXROLES - primary key on column RBACXUSERRBACXROLEKEY

Foreign Keys

None

Indexes

None

top

RBX_ROLE_ACEGI_ROLES Table

Structure

	RBX_ROLE_ACEGI_ROLES
PK	RBACXROLEACEGIROLEKEY
	ACEGIROLENAME RBACXROLEKEY

Indexes

Index	PK	Unique	Keys
PK_RBACXROLEACEGIROLES	✓		RBACXROLEACEGIROLEKEY

Description

Each Role Manager security role is mapped to an internal role/privileges, which defines the internal job functions in the tool interface. This mapping is stored in the RBX_ROLE_ACEGI_ROLES table. This table has a unique RBACXROLEACEGIROLEKEY and it references the RBX_ROLES table using the RBACXROLEKEY field.

Primary Keys

1.PK_RBACXROLEACEGIROLES - primary key on column RBACXROLEACEGIROLEKEY

Foreign Keys

None

Indexes

None

top

RBX_USER_RBX_ROLES_BU Table

Structure

	RBX_USER_RBX_ROLES_BU
PK	ID
	RBACXUSERKEY RBACXROLEKEY BUSINESSUNITKEY

Indexes

Index	PK	Unique	Keys
PK_RBX_USER_RBX_ROLES_BU	✓	Yes	ID
IX_RBX_USER_RBX_ROLES_BU_BID		No	BUSINESSUNITKEY
IX_RBX_USER_RBX_ROLES_BU_UID		No	RBACXUSERKEY
IX_RBX_USER_RBX_ROLES_BU_RID		No	RBACXROLEKEY
IX_RBX_USER_RBX_ROLES_BU		No	RBACXUSERKEY, RBACXROLEKEY, BUSINESSUNITKEY

Description

The association between a Role Manager user, a Role Manager security role, and a business unit is described in the RBX_USER_RBX_ROLES_BU table. A unique identifier ID identifies all the unique entries in the tables. There are references defined to the RBX_USERS, BUSINESSUNITS, and the RBX_ROLES tables.

Primary Keys

1.PK_RBX_USER_RBX_ROLES_BU - primary key on column ID

Foreign Keys

None

Indexes

1.IX_RBX_USER_RBX_ROLES_BU - non-unique index on column BUSINESSUNITKEY
2.IX_RBX_USER_RBX_ROLES_BU_USER - non-unique index on column RBACXUSERKEY
3.IX_RBX_USER_RBX_ROLES_BU_ROLE - non-unique index on column RBACXROLEKEY
4.IX_RBX_USER_RBX_ROLES_BU - non-unique composite index on columns RBACXUSERKEY, RBACXROLEKEY and BUSINESSUNITKEY

top

RBX_AUTHORITIES Table

Structure

	RBX_AUTHORITIES
	USERNAME ROLENAME

Indexes

Index	PK	Unique	Keys
IX_RBX_AUTHORITIES		No	USERNAME, ROLENAME

Description

Role Manager uses a security framework where, on logon, the user account is verified for the user's access levels by checking the user's list of roles. The RBX_AUTHORITIES table handles this framework. A username and the user's role are listed in this table and, if a user has multiple roles, multiple values are stored in the table for a single user. When the user logs in to Role Manager interface, the username-rolename association is looked up in this table and the user's access is granted.

Primary Keys

None

Foreign Keys

None

Indexes

1.IX_RBX_AUTHORITIES - non-unique composite index on columns USERNAME and ROLENAME

top

ACL_OBJECT_IDENTITY Table

Structure

	ACL_OBJECT_IDENTITY
PK	ID
	OBJECT_IDENTITY PARENT_OBJECT ACL_CLASS

Indexes

Index	PK	Unique	Keys
PK_ACL_OBJECT_IDENTITY	✓	Yes	ID

Description

The ACL_OBJECT_IDENTITY table defines the permissions assigned to each business unit or to the various components in Role Manager. The OBJECT_IDENTITY defines the permissions for a particular object. There can be a hierarchy associated in this table and this information is saved in the PARENT_OBJECT field.

Primary Keys

1.PK_ACL_OBJECT_IDENTITY - primary key on column ID

Foreign Keys

None

Indexes

None

top

ACL_PERMISSION Table

Structure

	ACL_PERMISSION
PK	ID
FK1,U1 U1	ACL_OBJECT_IDENTITY RECIPIENT MASK

Indexes

Index	PK	Unique	Keys
PK_ACL_PERMISSION	✓	Yes	ID
UNIQUE_RECIPIENT		Yes	ACL_OBJECT_IDENTITY, RECIPIENT

Description

The permissions on the objects defined in the previous tables are represented in the ACL_PERMISSION table. Each entry has a unique ID and an ACL_OBJECT_IDENTITY field reference to the ID field of the ACL_OBJECT_IDENTITY table. The RECIPIENT is the role or user for whom the permission is defined. The list of permissions are defined using integers where read/write access is defined using a certain number. Permissions are defined in much the same way that UNIX file permissions are defined.

Primary Keys

1.PK_ACL_PERMISSION - primary key on column ID

Foreign Keys

1.FK_ACL_PERMISSION_ACL_OBJ_ID - foreign key on column ACL_OBJECT_IDENTITY that references the ID field in the parent table ACL_OBJECT_IDENTITY. This foreign key is defined with CASCADE DELETE option.

Indexes

1.UNIQUE_RECIPIENT - composite unique constraint on columns ACL_OBJECT_IDENTITY and RECIPIENT

top

RBX_ACL_CLASS Table

Structure

	RBX_ACL_CLASS
PK	ID
U1	CLASS

Indexes

Index	PK	Unique	Keys
ACL_CLASS_PK	✓	Yes	ID
ACL_CLASS_UNIQUE_CLASS		Yes	CLASS

Description

The RBX_ACL_CLASS table is a part of the Role Manager security model. This table contains information related to the Java classes involved in defining security privileges within the system.

Primary Keys

1.ACL_CLASS_PK - primary key on column ID

Foreign Keys

None

Indexes

ACL_CLASS_UNIQUE_CLASS - unique index on column CLASS

top

RBX_ACL_ENTRY Table

Structure

	RBX_ACL_ENTRY
PK	ID
U1 , FK1 U1 FK2	ACL_OBJECT_IDENTITY ACE_ORDER SID MASK GRANTING AUDIT_SUCCESS AUDIT_FAILURE

Indexes

Index	PK	Unique	Keys
PK_RBX_ACL_ENTRY	✓	Yes	ID
ACL_ENTRY_UNIQUE_OID_ORDER		Yes	ACL_OBJECT_IDENTITY, ACE_ORDER
FK_ACL_ENTRY_AOI		No	ACL_OBJECT_IDENTITY
FK_ACL_ENTRY_ACL_SID		No	SID

Description

This table contains the Role Manager security objects as defined by the RBX_ACL_OBJECT_IDENTITY table. The RBX_ACL_ENTRY table is utilized in the creation of certification managers, enabling them to log in to the application in order to certify, revoke, or remediate a certificate.

Primary Keys

1.PK_RBX_ACL_ENTRY - primary key on column ID

Foreign Keys

1.FK_ACL_ENTRY_AOI - foreign key on column ACL_OBJECT_IDENTITY that references the ID field in the parent table RBX_ACL_OBJECT_IDENTITY.

2.FK_ACL_ENTRY_ACL_SID - foreign key on column SID that references the ID field in the parent table RBX_ACL_SID.

Indexes

1.ACL_ENTRY_UNIQUE_OID_ORDER - composite unique constraint on columns ACL_OBJECT_IDENTITY and ACE_ORDER.

top

RBX_ACL_OBJECT_IDENTITY Table

Structure

	RBX_ACL_OBJECT_IDENTITY
PK	ID
FK1,U1 U1 FK2 FK3	OBJECT_ID_CLASS OBJECT_ID_IDENTITY PARENT_OBJECT OWNER_SID ENTRIES_INHERITING

Indexes

Index	PK	Unique	Keys
ACL_O_I_PK	✓	Yes	ID
ACL_O_I_UNIQUE_CLASS_OID		Yes	OBJECT_ID_CLASS, OBJECT_ID_IDENTITY
FK_AOI_PARENT_OBJECT		No	PARENT_OBJECT
FK_AOI_ACL_CLASS		No	ID
FK_AOI_ACL_SID		No	OWNER_SID

Description

The RBX_ACL_OBJECT_IDENTITY table defines the privileges for different security objects made available to a RBX USER. The OBJECT_ID_CLASS column indicates the Java class used for the security definition, while the OBJECT_ID_IDENTITY column consists of the ID of the security object with which privileges are being associated.

Primary Keys

1.ACL_O_I_PK - primary key on column ID

Foreign Keys

1.FK_AOI_ACL_CLASS - foreign key on column OBJECT_ID_CLASS that references the ID field in the parent table RBX_ACL_CLASS.
2.FK_AOI_PARENT_OBJECT - foreign key on column PARENT_OBJECT that references the ID field in the parent table RBX_ACL_OBJECT_IDENTITY. This foreign key is defined with CASCADE and DELETE option.
3.FK_AOI_ACL_SID - foreign key on column OWNER_SID that references the ID in the parent table RBX_ACL_SID. This foreign key is defined with CASCADE and DELETE option.

Indexes

1.ACL_O_I_UNIQUE_CLASS_OID - composite unique constraint on columns OBJECT_ID_CLASS and OBJECT_ID_IDENTITY.

top

RBX_ACL_SID Table

Structure

	RBX_ACL_SID
PK	ID
	PRINCIPAL SID

Indexes

Index	PK	Unique	Keys
PK_RBX_ACL_SID	✓	Yes	ID

Description

The RBX_ACL_SID table represents a user, a principal, or an owner of a Role Manager security user account with whom security privileges are associated. These security privileges define the ability to add, remove, or update security objects.

Primary Keys

1. PK_RBX_ACL_SID - primary key on column ID

Foreign Keys

None

Indexes

None

top

RBX_AUDIT_ENTRIES Table

Structure

	RBX_AUDIT_ENTRIES
PK	RBACXAUDITENTRYKEY
	OBJECT_NAME DESCRIPTION USERNAME EXECUTIONDATE SERVICE OBJECT OBJECTKEY ACTIONPERFORMED LOCAL_ADDR REMOTE_ADDR REMOTE_HOST SERVER_NAME

Indexes

Index	PK	Unique	Keys
PK_RBX_AUDIT_ENTRIES	✓	Yes	RBACXAUDITENTRYKEY

Description

Role Manager maintains an audit log of all transactions. Each audit entry is stored as a separate record in the RBX_AUDIT_ENTRIES table. Every audit entry is defined by a unique RBACXAUDITENTRYKEY. In addition, the USERNAME, EXECUTIONDATE, DESCRIPTION, OBJECT_NAME, LOCAL_ADDR, REMOTE_ADDR, REMOTE_HOST, and SERVER_NAME fields show the logged audit details. Various services are used in Role Manager and these are listed under the SERVICE field in the table. The ACTIONPERFORMED field shows the type of action performed by the user.

Primary Keys

1.PK_RBX_AUDIT_ENTRIES - primary key on column RBACXAUDITENTRYKEY

Foreign Keys

None

Indexes

None

top

RBX_AUDIT_ENTRY_EXT_PROPS Table

Structure

	RBX_AUDIT_ENTRY_EXT_PROPS
PK	RBACXAUDITEXTENDEDPROPERTYKEY
I1	ENTITY ENTITYID ACTIONPERFORMED RBACXAUDITENTRYKEY PROPERTYNAME OLDVALUE NEWVALUE DESCRIPTION USERNAME ENTITY_NAME

Indexes

Index	PK	Unique	Keys
PK_RBX_AUDIT_ENTRY_EXT_PROPS	✓		RBACXAUDITEXTENDEDPROPERTYKEY
IX_RBX_AUDIT_ENTRY_EXT_PROPS		No	RBACXAUDITENTRYKEY

Description

Each audit entry has extended properties related to it. These properties are defined in the RBX_AUDIT_ENTRY_EXT_PROPS table. Each extended property has its entity defined and is associated with the RBACX_AUDIT_ENTRIES table using

the RBACXAUDITENTRYKEY.

Primary Keys

1.PK_RBX_AUDIT_ENTRY_EXT_PROPS - primary key on column RBACXAUDITEXTENDEDPROPERTYKEY

Foreign Keys

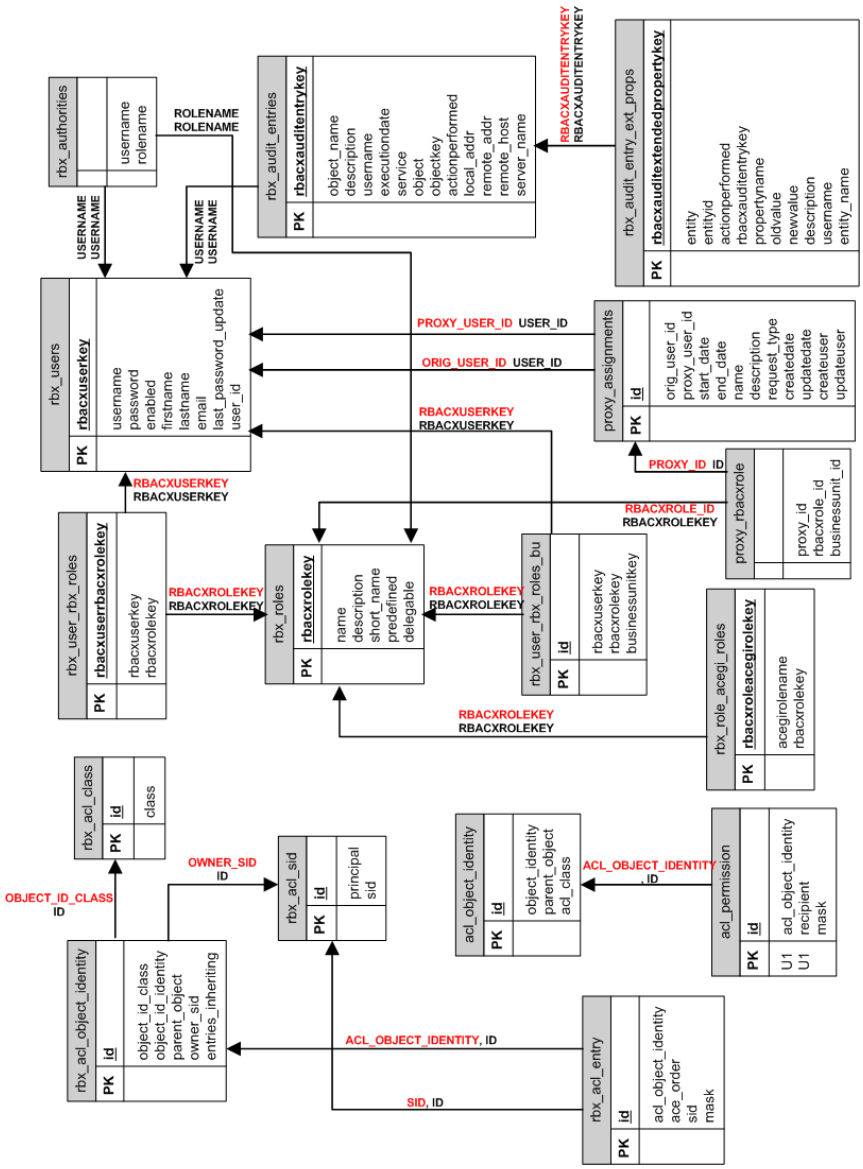
None

Indexes

1.IX_RBX_AUDIT_ENTRY_EXT_PROPS - non-unique index on column RBACXAUDITENTRYKEY

top

Security Module Entity Relationship



top

Miscellaneous Tables

The Miscellaneous Tables

This chapter describes the miscellaneous tables that are used in Role Manager.

CONFIGURATIONS Table

Structure

	CONFIGURATIONS
PK	CONFIGURATIONKEY
	CONFIGURATIONXML

Index

Index	PK	Unique	Keys
PK_CONFIGURATIONS	✓	Yes	CONFIGURATIONKEY

Description

Role Manager configuration details are defined in XML files, which are stored in the Role Manager database in the CONFIGURATIONS table. The CONFIGURATIONKEY column is a unique key that identifies the various configurations, while the CONFIGURATIONXML column points to the various configuration files that handle the connections to the provisioning system.

Primary Keys

1. PK_CONFIGURATIONS on column CONFIGURATIONKEY

Foreign Keys

None

Indexes

None

top

EMAIL_TEMPLATES Table

Structure

	EMAIL_TEMPLATES
PK	ID
	NAME DESCRIPTION SENDERNAME EMAILFROM EMAILTO EMAILCC EMAILBCC SUBJECT HTMLENABLED BODY REPORTS_ALLOWED CREATEUSER UPDATEUSER CREATEDATE UPDATEDATE

Index

Index	PK	Unique	Keys
PK_EMAIL_TEMPLATES	✓	Yes	ID

Description

The EMAIL_TEMPLATES table is used to store the email definitions utilized by different Role Manager modules. Each template is identified by a distinct ID. The REPORTS_ALLOWED field specifies whether the email would have a Role Manager report as an attachment. Additional information such as CREATEUSER, UPDATEUSER, CREATEDATE, and UPDATEDATE are provided for audit purposes.

Primary Keys

1. PK_EMAIL_TEMPLATES on column ID

Foreign Keys

None

Indexes

None

top

PASSWORD_DICTIONARY Table

Structure

	PASSWORD_DICTIONARY
PK	WORD

Index

Index	PK	Unique	Keys
PK_PASSWORD_DICTIONARY	✓	Yes	WORD

Description

Role Manager security supports user password validation. If enabled, one option checks the PASSWORD_DICTIONARY table for a list of prohibited words and, if a match is found, the password is not allowed. This validation is not case sensitive.

Primary Keys

1. PK_PASSWORD_DICTIONARY on column WORD

Foreign Keys

None

Indexes

None

top

SEQUENCES Table

Structure

	SEQUENCES
PK	SEQUENCENAME
	CURRENTVALUE

Index

Index	PK	Unique	Keys
PK_SEQUENCES	✓	Yes	SEQUENCENAME

Description

The SEQUENCES table holds the last IDs used for domain objects in the Role Manager tables. These IDs are stored in the CURRENTVALUE field.

Primary Keys

1. PK_SEQUENCES on column SEQUENCENAME

Foreign Keys

None

Indexes

None

top

PARAMETERS Table

Structure

	PARAMETERS
	PARAMETER_NAME INTEGER_VALUE LONG_VALUE DATE_VALUE BOOLEAN_VALUE TEXT_VALUE

Index

None

Description

The PARAMETERS table stores parameters of different types that are used by Role Manager. This table is used to store miscellaneous data.

Primary Keys

None

Foreign Keys

None

Indexes

None

top