

Service Warehouse Data Dictionary

Operations Center 5.0

September 30, 2011

Legal Notices

Novell, Inc., makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc., makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the [Novell International Trade Services Web page \(http://www.novell.com/info/exports/\)](http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell ass

umes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 1997-2012 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.
www.novell.com

Online Documentation

To access the latest online documentation for this and other Novell products, see the [Novell Documentation Web page \(http://www.novell.com/documentation/\)](http://www.novell.com/documentation/).

Novell Trademarks

For Novell trademarks, see the [Novell Trademark and Service Mark list \(http://www.novell.com/company/legal/trademarks/tmlist.html\)](http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Contents

About This Guide	5
1 Service Warehouse Overview	7
1.1 Data Capture for Performance and Service Level Data	8
1.2 Data Warehouse Performance Database Tables	9
1.3 Warehouse Database Tables	10
1.4 Normalizing Alarm History Data	10
2 Introduction to Databases	13
2.1 Conventions	13
2.2 Database Symbols	13
2.2.1 IBM DB2 Database Symbols	13
2.2.2 Microsoft SQL Server Database Symbols	14
2.2.3 Oracle Database Symbols	14
2.2.4 PostgreSQL Database Symbols	15
2.2.5 Sybase Database Symbols	15
3 Warehouse Database Tables	17
3.1 Database Entity Relationship Diagram	18
3.2 BSAAapters Table	19
3.3 BSAAAlarmData Table	19
3.4 BSAAAlarmElements Table	21
3.5 BSAAAlarmMetaData Table	22
3.6 BSAAAlarmProperties Table	22
3.7 BSAAAlarmQueue Table	22
3.8 BSAAAlarmSeries Table	23
3.9 BSAAAlarmValues Table	23
3.10 BSAAAlarmValueTypes Table	24
3.11 BSAConfigHistory Table	24
3.12 BSADataTypes Table	25
3.13 BSAElements Table	25
3.14 BSACFactSeriesData Table	26
3.15 BSACFactSeriesRootCauses Table	26
3.16 BSACFactSeriesSummary Table	27
3.17 BSACHosts Table	27
3.18 BSACRootCauseChains Table	28
3.19 BSACRootCauseHierarchy Table	28
3.20 BSACRootCauseReasons Table	28
3.21 BSACRootCauseTrees Table	29
3.22 BSACSeries Table	29
3.23 BSACTimeBand Table	30
3.24 BSACVersionWarehouse Table	31

About This Guide

The Novell® Operations Center *Service Warehouse Data Dictionary Guide* provides describes data dictionary. All database tables listed in this document can be accessed directly through third-party reporting tools, if a valid database driver such as ODBC or JDBC* is available from the database vendor.

Audience

This guide is intended for Operations Center system administrators that need to access Service Warehouse database tables via third-party reporting tools.

Guide Sections

- ♦ [Chapter 1, "Service Warehouse Overview," on page 7](#)
- ♦ [Chapter 2, "Introduction to Databases," on page 13](#)
- ♦ [Chapter 3, "Warehouse Database Tables," on page 17](#)

Additional Documentation

For other related documentation:

- ♦ [Novell Operations Center 5.0 Getting Started Guide](#)
- ♦ [Novell Operations Center 5.0 Server Installation Guide](#)
- ♦ [Novell Operations Center 5.0 Server Configuration Guide](#)
- ♦ [Novell Operations Center 5.0 User Guide](#)
- ♦ [Novell Operations Center 5.0 SQL Views Guide](#)

SQL Views also allows you to extract data from the Service Warehouse and use it in reports created through third-party tools. For more information about SQL Views, see the [Novell Operations Center 5.0 SQL Views Guide](#).

Documentation Updates

In general, Operations Center provides two main sources of information to help you configure and use Operations Center tools:

- ♦ **Online help:** The help is part of each product and available from multiple locations depending upon the product; look for help in the upper right of a window or dialog box as a drop-down menu option, a help button, or a question mark symbol (?).
- ♦ **Guides:** Available as PDF files on the installation CD, the guides provide an overview of how to solve a problem, complete a task, or use a specific Operations Center tool.

For the most recent version of the *Business Service Warehouse Data Dictionary*, visit the [Operations Center 5.0 Web site \(http://www.novell.com/documentation/noc50/\)](http://www.novell.com/documentation/noc50/). A password is required for access to the Operations Center documentation.

Documentation Conventions

In Novell documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path. The > symbol is also used to connect consecutive links in an element tree structure where you can either click a plus symbol (+) or double-click each element to expand them.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a forward slash to preserve case considerations in the UNIX* or Linux* operating systems.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation, or go to www.novell.com/documentation/feedback.html and enter your comments there.

1 Service Warehouse Overview

The Service Warehouse allows you to capture performance series, alarm history, and service level data through profiles that match one or more elements based on criteria defined by the administrator. Expressions defined for a profile determine the type of data that is captured, such as alarm severity, element conditions, or alarm fields.

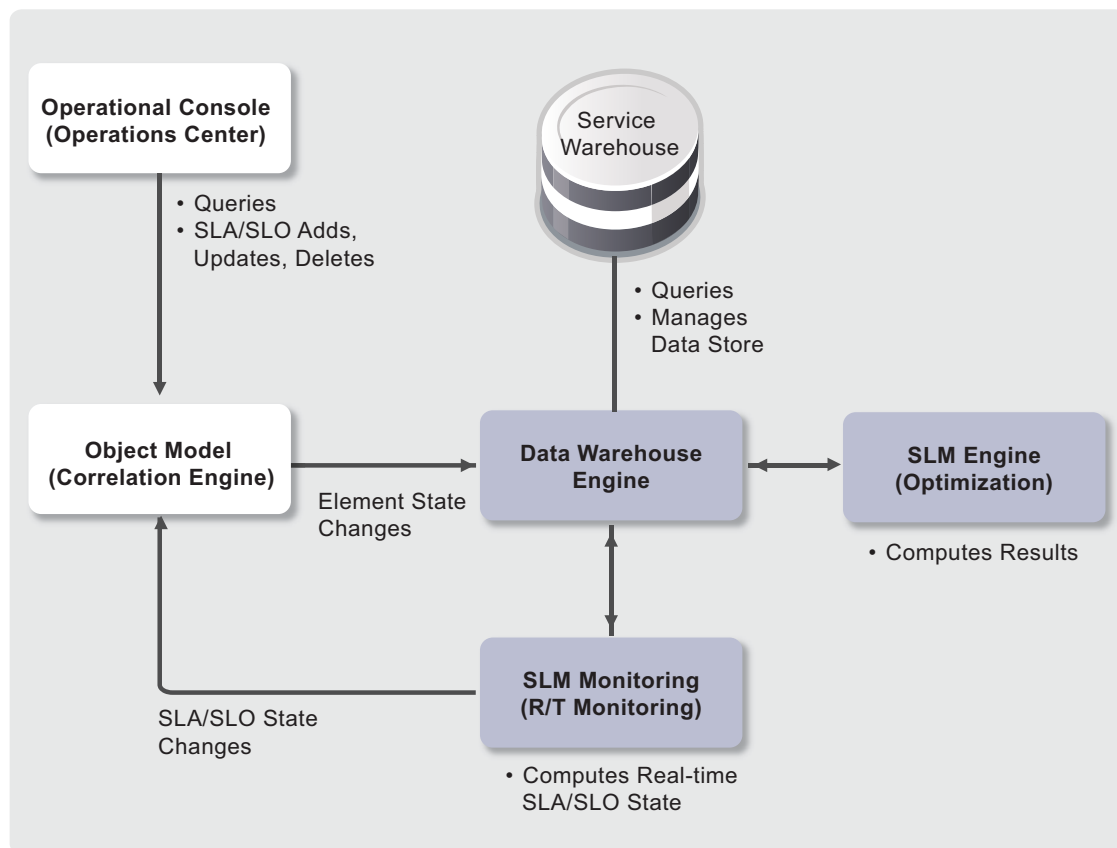
- ♦ [Section 1.1, “Data Capture for Performance and Service Level Data,” on page 8](#)
- ♦ [Section 1.2, “Data Warehouse Performance Database Tables,” on page 9](#)
- ♦ [Section 1.3, “Warehouse Database Tables,” on page 10](#)
- ♦ [Section 1.4, “Normalizing Alarm History Data,” on page 10](#)

1.1 Data Capture for Performance and Service Level Data

The Data Warehouse is an integral part of the ability of Novell® Operations Center to store alarm history and the ability of the Service Level Manager (SLM) to capture performance and service level information so that it can be exposed and analyzed later.

[Figure 1-1](#) provides an overview of how the Service Warehouse stores and feeds data for the Operations Center server:

Figure 1-1 Service Warehouse High-Level Architecture



The [Novell Operations Center 5.0 Server Configuration Guide](#) explains the process of setting up database definitions, Time Management objects (time categories, calendars, schedules), and Data Warehouse features (profiles, expressions, data warehouse settings) that are used by the Service Warehouse to store alarm history, performance, and service level information.

Basically, data is captured through profiles that match one or more elements based on criteria defined by the administrator. Expressions defined for a profile determine the type of data that is captured, such as alarm severity, element conditions, and alarm fields.

After a Operations Center administrator configures the data capture, users can view the information in various ways, depending upon the type of data.

Capturing and storing performance and service level data requires the purchase of the Service Level Manager. At purchase, you are given a specific license key to allow access to additional features for the purpose of capturing and analyzing historical performance data.

Performance data events are stored in one of two ways:

- ♦ **Scheduled Event:** The scheduler determines that a collection event should be fired, and an expression polls an element for specific information.
- ♦ **Non-Scheduled Event:** An event is received from a specific element and an expression (such as an Alarm expression or Alarm Field expression) is “matched” to the element.

1.2 Data Warehouse Performance Database Tables

At the first scheduled interval after all Time Management and Data Warehouse objects are set up and configured, the BSLM Scheduling Engine fires an event for BSLM expressions. This causes the engine to poll the matched elements for the specific performance data, then store them to the database tables, as described in [Table 1-1](#).

Table 1-1 Data Warehouse Performance Database Tables

Database Table	Description						
BSATimeBand	If the current time normalized to a five-second interval falls within an existing BSATimeBand record, the existing time band information is used. If not, then a new BSATimeBand record is created for the current performance event. The new/existing time band record is cached by the BSLM Engine to reduce database overhead for future events.						
BSASeries	The BSASeries table is queried to determine if a series record exists for the current Profile/Subject/Expression combination. For example: <table><tr><td>Profile</td><td>Test 1 Profile</td></tr><tr><td>Subject</td><td>TEC (that is, the Tec element above)</td></tr><tr><td>Expression</td><td>Element Conditions</td></tr></table> If a series record already exists, the existing series record is used to populate the foreign keys in the BSFactSeriesData record. If the series information does not exist, a new record is inserted and foreign keys populated as required. The new/existing BSASeries record is cached by the BSLM Engine to minimize database overhead.	Profile	Test 1 Profile	Subject	TEC (that is, the Tec element above)	Expression	Element Conditions
Profile	Test 1 Profile						
Subject	TEC (that is, the Tec element above)						
Expression	Element Conditions						
BSFactSeriesData	This table stores the actual performance data points retrieved by the BSLM Engine. The exception to this rule is the Alarm Field expression. Since alarms are inspected when the event is received by the Operations Center server, the specific alarm field is stored when processed.						
RootCauseTables:	If the expression (such as. ElementConditionChange) requires persistence of the root cause information. ♦ BSFactSeriesRootCauses ♦ BSARootCauseTrees ♦ BSARootCauseChains ♦ BSARootCauseHierarchy ♦ BSARootCauseReasons						

1.3 Warehouse Database Tables

When an alarm event is received, the event is forwarded to the BSLM Engine, and stored to the Warehouse database tables, as described in [Table 1-2](#).

Table 1-2 Warehouse Database Tables

Database Table	Description						
BSATimeBand	If the current time normalized to a five-second interval falls within an existing BSATimeBand record, the existing time band information is used. If not, then a new BSATimeBand record is created for the current performance event. The new/existing time band record is cached by the BSLM Engine to reduce database overhead for future events.						
BSASeries	The BSASeries table is queried to determine if a series record exists for the current Profile/Subject/Expression combination, with the exception of the Expression field. The BSASeries record for an alarm event in the sample setup contains the following data: <table><tr><td>Profile</td><td>Test 1 Profile</td></tr><tr><td>Subject</td><td>TEC (that is, the Tec element above)</td></tr><tr><td>Expression</td><td>Alarms</td></tr></table> If a series record already exists, the existing series record is used to populate the foreign keys in the BSAAAlarmData record. The new/existing BSASeries record is cached by the BSLM Engine to minimize database overhead.	Profile	Test 1 Profile	Subject	TEC (that is, the Tec element above)	Expression	Alarms
Profile	Test 1 Profile						
Subject	TEC (that is, the Tec element above)						
Expression	Alarms						
BSAElements	The BSLM Engine then checks the alarm event to determine all affected elements. Each affected element is checked against the BSAElements table to confirm their existence. If an entry for the Element DName is already present in this table, it is retrieved; otherwise a new record is inserted. Both new and existing records are cached by the BSLM Engine to reduce database overhead.						
BSAAAlarmData	Normalized alarm data from the event is then broken out and set in the appropriate fields of the BSAAAlarmData table. Also, the entire alarm object is serialized and stored to the BSAAAlarmData.AlarmBlob field.						
BSAAAlarmElements	After all affected elements have been identified and appropriate BSAElements records are cached, the next step is to create reference entries between the BSAAAlarmData and BSAElements table. This is done by inserting a new row for each affected ElementKey and the current AlarmKey into the BSAAAlarmElements table.						
BSAAAlarmProperties	Specific alarm properties are persisted to BSAAAlarmProperties to identify alarm hash values for deduplication purposes, and status flags for general alarm work flow.						

1.4 Normalizing Alarm History Data

The Normalize Alarm History Data feature can be set on individual Alarm expression instances attached to your profiles. This option causes the BSLM Engine to break out the alarm fields into normalized tables, which can then be accessed by third-party reporting packages. To reduce

overhead on the Operations Center server, alarm breakout events are first queued to the BSAAAlarmQueue table, then processed in batches by a lower priority maintenance thread to maximize database throughput.

Alarms queued to the BSAAAlarmQueue table are retrieved in a first-in/first-out order. Alarm fields specified in the Alarm expression are extracted from the serialized object (BSAAAlarmData.AlarmBlob), and stored to the BSAAAlarmXML table. One row is added per alarm and contains an XML representation of the alarm.

2 Introduction to Databases

Although other open source or public domain databases might work, only the databases listed in the [Novell Operations Center 5.0 Getting Started Guide](#) have been certified by Operations Center for production use.

This section defines all the databases supported by the Novell® Operations Center Service Warehouse.

- ♦ [Section 2.1, “Conventions,” on page 13](#)
- ♦ [Section 2.2, “Database Symbols,” on page 13](#)

2.1 Conventions

The following conventions are imposed on the database schema to maintain compatibility between the supported database environments:

- ♦ All date/time fields are represented as a long data type. The value of the date/time field is the time in milliseconds since January 1, 1970 (UNIX EPOCH time).
- ♦ Numeric fields use the NUMBER data type, which is supported by all certified databases.
- ♦ Boolean values are stored as a SMALLINT data type where 0 is false and 1 is true. It is the responsibility of the application code to convert to the proper data type after the data has been retrieved.
- ♦ All binary data is stored as a LONG RAW data type because all databases do not support the BLOB and CLOB native data types.

2.2 Database Symbols

To maintain application code compatibility among the supported databases, some data types are referenced as symbols in the Warehouse schema script to deal with particular limitations on specific database environments. Symbols are represented as `$$SYMBOL_NAME$` in the database tables. The following tables list each symbol and its redefined data type:

- ♦ [Section 2.2.1, “IBM DB2 Database Symbols,” on page 13](#)
- ♦ [Section 2.2.2, “Microsoft SQL Server Database Symbols,” on page 14](#)
- ♦ [Section 2.2.3, “Oracle Database Symbols,” on page 14](#)
- ♦ [Section 2.2.4, “PostgreSQL Database Symbols,” on page 15](#)
- ♦ [Section 2.2.5, “Sybase Database Symbols,” on page 15](#)

2.2.1 IBM DB2 Database Symbols

[Table 2-1](#) describes IBM DB2 database symbols.

Table 2-1 IBM DB2 Database Symbols

Symbol Name	Data Type Value	Description
STRING	VARCHAR(3000)	Large string value
IDX_STRING	VARCHAR(254)	Maximum string value for fields included in an index
VARCHAR	VARCHAR	Not redefined
ID	NUMERIC(24)	Unique key identifier (represented as a LONG value)
LONG	NUMERIC(24)	Long value
DOUBLE	DOUBLE	Not redefined
BOOLEAN	SMALLINT	Boolean value
TIMEPOINT	NUMERIC(32)	Time point value represented as a LONG value
DATETIME	TIMESTAMP	A native Date/Time field

2.2.2 Microsoft SQL Server Database Symbols

[Table 2-2](#) describes Microsoft SQL server database symbols.

Table 2-2 Microsoft SQL Server Database Symbols

Symbol Name	Data Type Value	Description
STRING	VARCHAR(254)	Large string value
IDX_STRING	VARCHAR(254)	Maximum string value for fields included in an index
VARCHAR	VARCHAR	Not redefined
ID	NUMERIC(24)	Unique key identifier (represented as a LONG value)
LONG	NUMERIC(24)	Long value
DOUBLE	MONEY	Double value
BOOLEAN	SMALLINT	Boolean value
TIMEPOINT	NUMERIC(32)	Time point value represented as a LONG value
DATETIME	DATETIME	A native Date/Time field

2.2.3 Oracle Database Symbols

[Table 2-3](#) describes Oracle database symbols.

Table 2-3 Oracle Database Symbols

Symbol Name	Data Type Value	Description
STRING	VARCHAR2(3000)	Large string value
IDX_STRING	VARCHAR2(800)	Maximum string value for fields included in an index

Symbol Name	Data Type Value	Description
VARCHAR	VARCHAR2	Redefined to use a more efficient variable length character field
ID	NUMERIC(32)	Unique key identifier (represented as a LONG value)
LONG	NUMERIC(32)	Long value
DOUBLE	DOUBLE PRECISION	Double value
BOOLEAN	SMALLINT	Boolean value
TIMEPOINT	NUMERIC(32)	Time point value represented as a LONG value
DATETIME	DATE	A native Date/Time field

2.2.4 PostgreSQL Database Symbols

[Table 2-4](#) describes PostgreSQL database symbols.

Table 2-4 *PostgreSQL Database Symbols*

Symbol Name	Data Type Value	Description
STRING	TEXT	Text value
IDX_STRING	VARCHAR(254)	Maximum string value for fields included in an index
VARCHAR	VARCHAR	Not redefined
ID	NUMERIC(24)	Unique key identifier (represented as a LONG value)
LONG	NUMERIC(24)	Long value
DOUBLE	DOUBLE	Double value
BOOLEAN	SMALLINT	Boolean value
TIMEPOINT	NUMERIC(32)	Time point value represented as a LONG value
DATETIME	TIMESTAMP	A native Date/Time field

2.2.5 Sybase Database Symbols

[Table 2-5](#) describes Sybase database symbols.

Table 2-5 *Sybase Database Symbols*

Symbol Name	Data Type Value	Description
STRING	TEXT	Text value
IDX_STRING	VARCHAR(254)	Maximum string value for fields included in an index
VARCHAR	VARCHAR	Not redefined
ID	NUMERIC(24)	Unique key identifier (represented as a LONG value)
LONG	NUMERIC(24)	Long value

Symbol Name	Data Type Value	Description
DOUBLE	DOUBLE	Double value (not redefined for sybase)
BOOLEAN	SMALLINT	Boolean value
TIMEPOINT	NUMERIC(32)	Time point value represented as a LONG value
DATETIME	DATETIME	A native Date/Time field

3 Warehouse Database Tables

This section provides descriptions and table columns for all Service Warehouse database tables.

For all tables in this section, the following column heading abbreviations are used:

P = Primary Key

M = Mandatory

F = Foreign

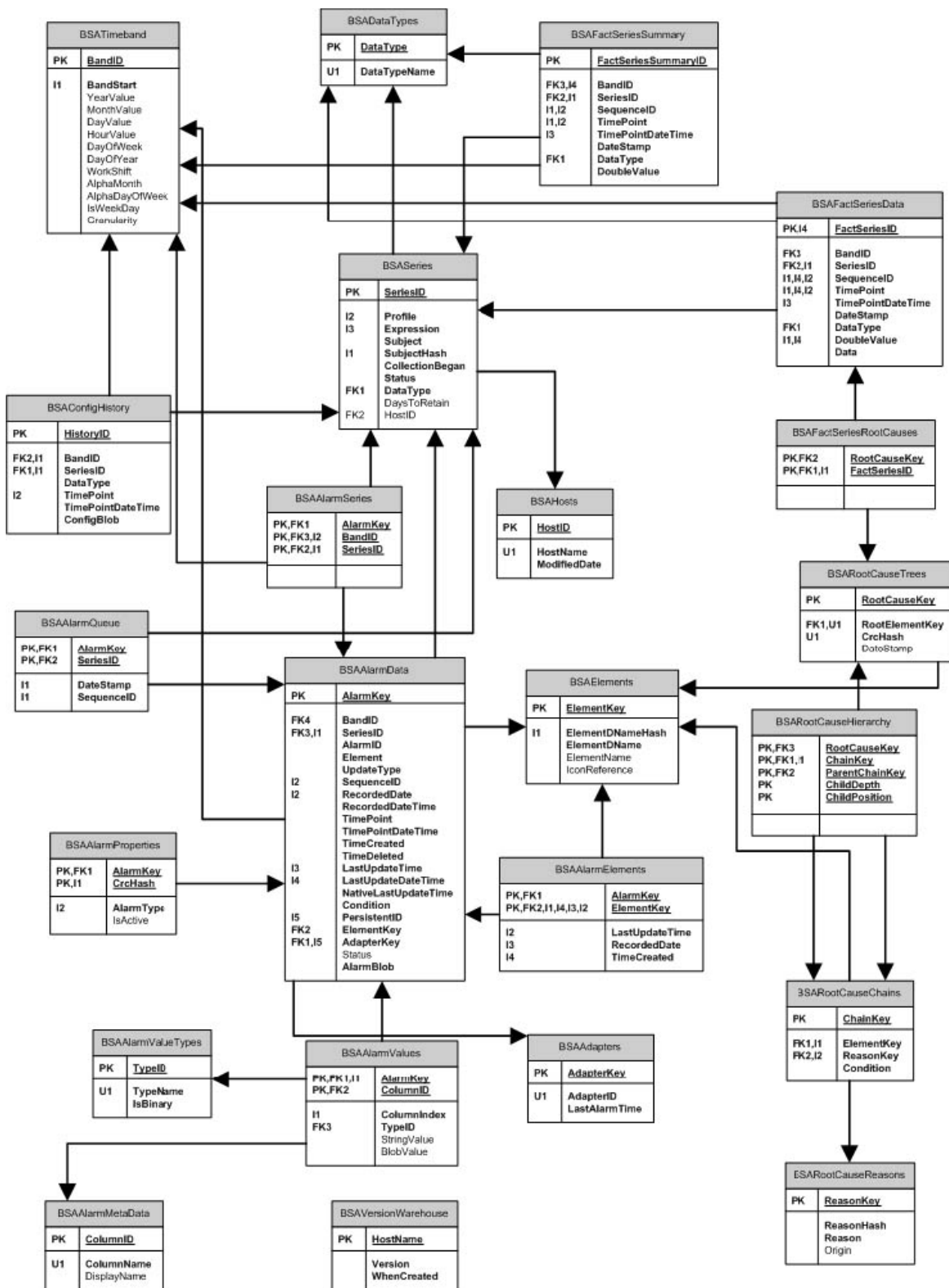
The following sections contain the entity relationship diagram for the Service Warehouse database, as well as a section for all warehouse tables:

- [Section 3.1, "Database Entity Relationship Diagram," on page 18](#)
- [Section 3.2, "BSAAdapters Table," on page 19](#)
- [Section 3.3, "BSAAlarmData Table," on page 19](#)
- [Section 3.4, "BSAAlarmElements Table," on page 21](#)
- [Section 3.5, "BSAAlarmMetaData Table," on page 22](#)
- [Section 3.6, "BSAAlarmProperties Table," on page 22](#)
- [Section 3.7, "BSAAlarmQueue Table," on page 22](#)
- [Section 3.8, "BSAAlarmSeries Table," on page 23](#)
- [Section 3.9, "BSAAlarmValues Table," on page 23](#)
- [Section 3.10, "BSAAlarmValueTypes Table," on page 24](#)
- [Section 3.11, "BSAConfigHistory Table," on page 24](#)
- [Section 3.12, "BSADataTypes Table," on page 25](#)
- [Section 3.13, "BSAElements Table," on page 25](#)
- [Section 3.14, "BSAFactSeriesData Table," on page 26](#)
- [Section 3.15, "BSAFactSeriesRootCauses Table," on page 26](#)
- [Section 3.16, "BSAFactSeriesSummary Table," on page 27](#)
- [Section 3.17, "BSAHosts Table," on page 27](#)
- [Section 3.18, "BSARootCauseChains Table," on page 28](#)
- [Section 3.19, "BSARootCauseHierarchy Table," on page 28](#)
- [Section 3.20, "BSARootCauseReasons Table," on page 28](#)
- [Section 3.21, "BSARootCauseTrees Table," on page 29](#)
- [Section 3.22, "BSASeries Table," on page 29](#)
- [Section 3.23, "BSATimeBand Table," on page 30](#)
- [Section 3.24, "BSAVersionWarehouse Table," on page 31](#)

3.1 Database Entity Relationship Diagram

The following diagram shows the relationships among the various database tables:

Figure 3-1 *Table Relationships*



3.2 BSAAapters Table

The BSAAapters table contains information pertaining to adapters that have reported alarms to the event store facility. Only adapters that have stored alarms as a result of an active profile appear in it.

Table 3-1 BSAAapters Table

Name	Type	P	M	F	Description
Adapter Key	\$ID\$	Y	Y	N	Random number generated at row creation time to identify the adapter DName.
AdapterID	\$IDX_STRING\$	N	Y	N	DName of the Adapter Administration element.
LastAlarmTime	NUMBER	N	Y	N	The last time the Operations Center server received an alarm refresh packet for the current adapter. This allows the Operations Center server to discard duplicate refresh packets when restarted.

3.3 BSAAalarmData Table

All alarm events are stored in the BSAAalarmData table if an appropriate profile is present. This table contains the common set of normalized alarm fields and the entire contents of the alarm in binary form.

Table 3-2 BSAAalarmData Table

Name	Type	P	M	F	Description
AlarmKey	\$ID\$	Y	Y	N	Unique random number generated when the alarm record was stored to the database.
BandID	\$ID\$	N	Y	Y	Foreign key reference to the BSATimeBand table.
SeriesID	\$ID\$	N	Y	Y	Foreign key to the BSASeries table.
AlarmID	NUMBER	N	Y	N	Nonpersistent Alarm ID generated by the network management system.
Element	NUMBER	N	Y	N	The ID of the affected element. This ID is not unique and might be reassigned by the Operations Center server after a server restart.
UpdateType	NUMBER	N	Y	N	Alarm update type. Supported types include: 0=Alarm Created 1=Alarm updated 2=Alarm Deleted
RecordedDate	\$TIMEPOINT\$	N	Y	N	Date and time that the BSAAalarmData record was created (represented in milliseconds).
RecordedDateTime	\$DATETIME\$	N	Y	N	Date and time that the BSAAalarmData record was created (represented as a native date/time object).
SequenceID	LONG	N	Y	N	Maintains data order when the data occurs within the same millisecond.

Name	Type	P	M	F	Description
TimePoint	\$TIMEPOINT\$	N	Y	N	The time point of the alarm record. This field is set to the date and time when the Operations Center server received the alarm (represented in milliseconds).
TimePointDateTime	\$DATETIME\$	N	Y	N	The time point of the alarm record. This field is set to the date and time when the Operations Center server received the alarm (represented in milliseconds).
TimeCreated	\$TIMEPOINT\$	N	Y	N	The date and time the alarm was created. This field is the same as NativeLastUpdateTime for alarm create records. However, if this is an alarm delete entry, the TimeCreated field is set to Long.MAX_VALUE; that is, 9223372036854775807.
TimeDeleted	\$TIMEPOINT\$	N	Y	N	The date and time the alarm was closed or deleted. This field is the same date as NativeLastUpdateTime for alarm delete records. However, if this is an alarm create entry, this field is set to Long.MAX_VALUE; that is, 9223372036854775807.
LastUpdateTime	\$TIMEPOINT\$	N	Y	N	The date and time of the most recent update of the current alarm. This is a skewed date/time, meaning the time is adjusted for differences between the network management system and the Operations Center server. This value is represented in milliseconds.
LastUpdateDateTime	\$DATETIME\$	N	Y	N	The date and time of the most recent update of the current alarm. This is a skewed date/time; the time is adjusted for differences between the network management system and the Operations Center server. This value is represented as a native date/time object.
NativeLastUpdateTime	\$TIMEPOINT\$	N	Y	N	The most recent update time of the current alarm. This is the nonskewed date/time, meaning no adjustments have been made between the network management system and Operations Center server time.
Condition	NUMBER	N	Y	N	Element condition for current alarm. Current values include: 0=Unknown 1=Critical 2=Major 3=Minor 4=Informational 5=OK 6=Initial 7=Usage Idle 8=Usage Active 9=Usage Busy

Name	Type	P	M	F	Description
PersistentID	\$IDX_STRING\$	N	Y	N	A unique persistent ID associated with the current alarm. This ID is persistent across Operations Center server restarts, and is unique to each adapter.
ElementKe	\$ID\$	N	Y	Y	Foreign key to the BSAElements table.
AdapterKe	\$ID\$	N	Y	Y	Foreign key reference to the Adapters table.
Status	\$VARCHAR\$(32)	N	N	N	Normalized status of the alarm. Current values include: OPEN ACK SUSPENDED CLOSED DELETED
AlarmBlob	LONG RAW	N	N	N	Alarm data stored as a serialized object. Cannot be viewed by external reporting tools.

3.4 BSAAAlarmElements Table

The BSAAAlarmElements table contains a list of all affected elements for a particular alarm. It is used mainly as a reference table to join the [BSAAAlarmData](#) and [BSAElements](#) tables during alarm history queries.

Table 3-3 BSAAAlarmElements Table

Name	Type	P	M	F	Description
AlarmKey	\$ID\$	Y	Y	Y	Foreign key reference to the BSAAAlarmData table.
ElementKey	\$ID\$	Y	Y	Y	Foreign key to the BSAElements table.
RecordedDate	\$TIMEPOINT\$	N	Y	N	Date and time that the BSAAAlarmData record was created (represented in milliseconds).
TimeCreated	\$TIMEPOINT\$	N	Y	N	The date and time the alarm was created. This field is the same as NativeLastUpdateTime for alarm create records. However, if this is an alarm delete entry, the TimeCreated field is set to Long.MAX_VALUE; that is, 9223372036854775807.
LastUpdateTime	\$TIMEPOINT\$	N	Y	N	The date and time of the most recent update of the current alarm. This is a skewed date/time, meaning the time is adjusted for differences between the network management system and the Operations Center server. This value is represented in milliseconds.
AlarmType	CHAR(1)	N	Y	N	Alarm type.

3.5 BSAAAlarmMetaData Table

Alarm metadata for the event store breakout feature is stored in the `BSAAAlarmMetaData` table. Data is only present if the Normalize feature is enabled for the alarm history expression on the profile.

Table 3-4 *BSAAAlarmMetaData Table*

Name	Type	P	M	F	Description
ColumnID	\$ID\$	Y	Y	N	Unique random number generated when the alarm field metadata was created.
ColumnName	VARCHAR2(32)	N	Y	N	Alarm column name.
DisplayName	VARCHAR2(64)	N	N	N	The alarm field display name. Usually equals the alarm field name, but might be different for some adapters.

3.6 BSAAAlarmProperties Table

The `BSAAAlarmProperties` table contains basic information regarding an alarm record in the [BSAAAlarmData](#) table. This includes an alarm type used for channel support, a 32-bit CRC hash used for alarm deduplication, and a flag to determine if the alarm is considered active.

Table 3-5 *BSAAAlarmProperties Table*

Name	Type	P	M	F	Description
AlarmKey	\$ID\$	Y	Y	Y	Foreign key reference to the BSAAAlarmData table.
AlarmType	CHAR(1)	N	Y	N	Alarm type.
CrcHash	INTEGER	Y	Y	N	32-bit CRC hash computed on key alarm fields.
IsActive	\$BOOLEAN\$	N	N	N	Flag to indicate if the current alarm is active. (Currently unused).

3.7 BSAAAlarmQueue Table

The `BSAAAlarmQueue` table is used internally by the Warehouse to queue alarm records for normalization. Alarm history data normalization occurs as a lower priority maintenance thread to prevent performance bottlenecks during normal operations.

Table 3-6 *BSAAAlarmQueue Table*

Name	Type	P	M	F	Description
AlarmKey	\$ID\$	Y	Y	Y	Foreign key reference to the BSAAAlarmData table.
SeriesID	\$ID\$	Y	Y	Y	Foreign key reference to the BSASeries table.
DateStamp	\$TIMEPOINT\$	N	Y	N	The date and time that the BSAAAlarmQueue record was created.

Name	Type	P	M	F	Description
SequenceID	LONG	N	Y	N	Maintains data order when the data occurs within the same millisecond.

3.8 BSAAAlarmSeries Table

The BSAAAlarmSeries table links each alarm record to the associated performance series and time band records.

Table 3-7 BSAAAlarmSeries Table

Name	Type	P	M	F	Description
Alarm Key	\$ID\$	Y	Y	Y	Foreign key reference to the BSAAAlarmData table.
BandID	\$ID\$	Y	Y	Y	Foreign key reference to the BSATimeband table.
SeriesID	\$ID\$	Y	Y	Y	Foreign key reference to the BSASeries table.

3.9 BSAAAlarmValues Table

Normalized alarm history data field values are stored in the BSAAAlarmValues table when the Normalize feature is enabled for the Alarms history expression on a profile. The table contains one row for each field associated with an alarm.

Table 3-8 BSAAAlarmValues Table

Name	Type	P	M	F	Description
AlarmKey	\$ID\$	Y	Y	Y	Foreign key reference to the BSAAAlarmData table.
ColumnID	\$ID\$	Y	Y	Y	Foreign key reference to the BSAMetaData table.
ColumnIndex	NUMBER	N	Y	N	Index of the current field within the alarm.
TypeID	NUMBER	N	Y	Y	Foreign key to the BSAAAlarmValueTypes table.
StringValue	\$STRING\$	N	N	N	The string representation of the alarm field.
BlobValue	\$BLOB\$	N	N	N	Alarm field value for non-ASCII data.

3.10 BSAAAlarmValueTypes Table

Each alarm might have different CORBA data types associated with each field. The BSAAAlarmValueTypes table stores the different CORBA data types for reference purposes during alarm history data normalization.

Table 3-9 BSAAAlarmValueTypes Table

Name	Type	P	M	F	Description
TypeID	\$ID\$	Y	Y	N	The internal CORBA data type identifier for the current alarm field. This field is used to convert the binary form of the current field to its string representation.
TypeName	VARCHAR2(32)	N	Y	N	The string representation of the current internal CORBA identifier.
IsBinary	\$BOOLEAN\$	N	Y	N	Boolean flag to indicate that the alarm field value contains binary data.

3.11 BSAConfigHistory Table

The BSAConfigHistory table persists Service Level Manager (SLM) configuration information when modified. The table is currently only used for historical auditing purposes. Configuration information stored to it includes Service Level Agreements and time management data.

Table 3-10 BSAConfigHistory Table

Name	Type	P	M	F	Description
HistoryID	\$ID\$	Y	Y	N	A unique random number generated when the record is created.
BandID	\$ID\$	N	Y	Y	Foreign key to the BSATimeband table.
SeriesID	\$ID\$	N	Y	Y	Foreign key to the BSASeries table.
DataType	INTEGER	N	Y	N	Configuration history data type ID. A sequential number representing the data type.
TimePoint	\$LONG\$	N	Y	N	Date and time the configuration information record was recorded.
TimePointDateTime	\$DATETIME\$	N	Y	N	Date and time the configuration information record was recorded. This value is stored in the database native DATE format.
ConfigBlob	\$BLOB\$	N	Y	N	The configuration information stored as a serialized object.

3.12 BSADataTypes Table

The BSADataTypes table contains a list of data types supported by the BSLM Engine. All performance tables contain a foreign key constraint to this table to maintain referential integrity.

Table 3-11 BSADataTypes Table

Name	Type	P	M	F	Description
DataType	NUMBER	Y	Y	N	Performance information data type ID. A sequential number representing the data type.
DataTypeName	\$IDX_STRING\$	N	Y	N	Performance information data type. Supported data types include: Boolean Byte Short Integer Long Float Double String Object ConditionCounts Element Condition BSAAlarm

3.13 BSAElements Table

The BSAElements table contains a list of elements from which the BSLM Engine has received alarms. The table is used to normalize element DNames and is referenced by other BSLM Event store tables.

Table 3-12 BSAElements Table

Name	Type	P	M	F	Description
ElementKey	\$ID\$	Y	Y	N	A unique random number generated for the element when the record was created.
ElementDName	\$IDX_STRING\$	N	Y	N	The DName of the element for the current record. This field is URL encoded.
ElementName	\$STRING\$	N	N	N	The short name of the element for the current record.
IconReference	\$STRING\$	N	N	N	The icon associated to the element for the current record.
ElementDNameHash	NUMBER	N	Y	N	A 32-bit CRC hash of the ElementDName field. Used for faster searches.

3.14 BSFactSeriesData Table

The BSFactSeriesData table is the main Service Warehouse data repository. It contains all SLM performance information and is used as the primary data repository for graphing and charting performance data.

Table 3-13 BSFactSeriesData Table

Name	Type	P	M	F	Description
FactSeriesID	\$ID\$	Y	Y	N	A unique random number generated when the performance record was created.
BandID	\$ID\$	N	Y	Y	Foreign key reference to the BSATimeBand table.
SequenceID	LONG	N	Y	N	Maintains data order when the data occurs within the same millisecond.
SeriesID	\$ID\$	N	Y	Y	Foreign key to the BSASeries table.
TimePoint	\$TIMEPOINT\$	N	Y	N	The time point of the performance series record. This field is set to the date and time when the Operations Center server received the performance event. This value is represented in milliseconds.
TimePointDateTime	\$DATETIME\$	N	Y	N	The time point of the performance series record. This field is set to the date and time when the Operations Center server received the performance event. This value is represented as a native date/time object.
DateStamp	\$DATETIME\$	N	Y	N	Date and time that the BSFactSeriesData record was created.
DataType	NUMBER	N	Y	Y	Foreign key reference to the BSADataTypes table.
DoubleValue	\$DOUBLE\$	N	Y	N	If the performance data can be represented as a numeric value, it is converted to a double and inserted to this field. This provides an easy mechanism to produce averages or sums.
Data	VARCHAR2(128)	N	Y	N	Performance information data represented in string form. This data is converted to the proper data type by the Warehouse repository. All numeric data is stored as a string.

3.15 BSFactSeriesRootCauses Table

The BSFactSeriesRootCauses table is the summary table that links root cause information back to a performance series.

Table 3-14 BSFactSeriesRootCauses Table

Name	Type	P	M	F	Description
RootCauseKey	\$ID\$	Y	Y	Y	Foreign key reference to the BSARootCauseTrees table.

Name	Type	P	M	F	Description
FactSeriesID	\$ID\$	Y	Y	Y	Foreign key reference to the BSAFactSeriesData table.

3.16 BSAFactSeriesSummary Table

The BSAFactSeriesSummary table is a summary performance data repository. It contains a summary of BSLM performance information.

Table 3-15 BSAFactSeriesSummary Table

Name	Type	P	M	F	Description
FactSeriesSummaryID	\$ID\$	Y	Y	N	A unique random number generated when the performance record was created.
BandID	\$ID\$	N	Y	Y	Foreign key reference to the BSATimeBand table.
SeriesID	\$ID\$	N	Y	Y	Foreign key to the BSASeries table.
SequenceID	LONG	N	Y	N	Maintains data order when the data occurs within the same millisecond.
TimePoint	\$TIMEPOINT\$	N	Y	N	The summary time point of the performance series record.
TimePointDateTime	\$DATETIME\$	N	Y	N	The summary time point of the performance series record. This value is represented as a native date/time object.
DateStamp	\$DATETIME\$	N	Y	N	Date and time that the BSAFactSeriesSummary record was created.
DataType	NUMBER	N	Y	Y	Foreign key reference to the BSADataTypes table.
DoubleValue	\$DOUBLE\$	N	Y	N	If the performance data can be represented as a numeric value, it is converted to a double and inserted to this field. This provides an easy mechanism to produce averages or sums.

3.17 BSAHosts Table

The BSAHosts table contains information pertaining to the Operations Center server persisting data to the Service Warehouse database.

Table 3-16 BSAHosts Table

Name	Type	P	M	F	Description
HostID	\$ID\$	Y	Y	N	Random number generated at row creation time to identify the host.
HostName	\$IDX_STRING\$	N	Y	N	The host name of the system persisting data to the Service Warehouse database.

Name	Type	P	M	F	Description
ModifiedDate	\$TIMEPOINT\$	N	Y	N	The last time the BSLM Engine received an event (currently not in use).

3.18 BSARootCauseChains Table

The BSARootCauseChains table contains information regarding the parent element that is being affected by an element's change in condition, with a short description about why it is affected.

Table 3-17 BSARootCauseChains Table

Name	Type	P	M	F	Description
ChainKey	\$ID\$	Y	Y	N	Random number generated at row creation time to identify the Root Cause chain.
ElementKey	\$ID\$	N	Y	Y	Foreign key to the BSAElements table.
ReasonKey	\$ID\$	N	Y	Y	Foreign key to the BSARootCauseReasons table.
Condition	INTEGER	N	Y	N	The condition of the element when the root cause information was retrieved.

3.19 BSARootCauseHierarchy Table

The BSARootCauseHierarchy table contains information pertaining to the root cause hierarchy that is affected by the condition of an element.

Table 3-18 BSARootCauseHierarchyTable

Name	Type	P	M	F	Description
RootCauseKey	\$ID\$	Y	Y	Y	Foreign key to the BSARootCauseTrees table.
ChainKey	\$ID\$	Y	Y	Y	Foreign key to the BSARootCauseChains table.
ParentChainKey	\$ID\$	Y	Y	Y	Foreign key to the BSARootCauseChains table.
ChildDepth	INTEGER	Y	Y	N	The child depth in the hierarchy.
ChildPosition	INTEGER	Y	Y	N	The child position in the hierarchy.

3.20 BSARootCauseReasons Table

The BSARootCauseReasons table contains a list of reasons or descriptions for root cause situations.

Table 3-19 BSARootCauseReasons Table

Name	Type	P	M	F	Description
ReasonKey	\$ID\$	Y	Y	N	Random number generated at row creation time to identify the reason record.

Name	Type	P	M	F	Description
ReasonHash	INTEGER	N	Y	N	A 32-bit CRC hash of the reason information. This value is used to identify duplicate root cause reasons.
Reason	\$STRING\$	N	Y	N	A description of the root cause occurrence.
Origin	\$STRING\$	N	N	N	The origin of the root cause event.

3.21 BSARootCauseTrees Table

The BSARootCauseTrees table stores the element and time when a condition change affected the element.

Table 3-20 BSARootCauseTrees Table

Name	Type	P	M	F	Description
RootCauseKey	\$ID\$	Y	Y	N	Random number generated at row creation time to identify the root cause tree record.
RootElementKey	\$ID\$	N	Y	Y	Foreign key reference to the BSAElements table.
CrcHash	INTEGER	N	Y	N	A 32-bit CRC hash of the entire root cause structure. This value is used to identify and deduplicate root cause events.
DateStamp	\$DATETIME\$	N	N	N	Date and time that the BSARootCauseTrees record was created.

3.22 BSASeries Table

The BSASeries table contains all series information related to configured profiles. Each profile has separate entries for the various *profile/subject(DName)/expression* combination.

Table 3-21 BSASeries Table

Name	Type	P	M	F	Description
SeriesID	\$ID\$	Y	Y	N	A unique random number generated at performance series creation time.
Subject	\$IDX_STRING\$	N	Y	N	The subject of the current profile record. This is usually the DName of the element for which performance data is collected.
SubjectHash	NUMBER	N	Y	N	A 32-bit CRC hash of the Subject field; used for faster searches.
Profile	\$IDX_STRING\$	N	Y	N	The performance series profile name. This name is assigned by the user at profile creation time.
Expression	\$IDX_STRING\$	N	Y	N	The expression name associated with the current performance series.

Name	Type	P	M	F	Description
CollectionBegan	\$DATETIME\$	N	Y	N	Date and time when the profile began collecting data.
Status	CHAR(1)	N	Y	N	Performance series status. Current values include A=Active, D=Deleted. If the series record is in a deleted state, all data associated with the current series is purged from the database at the next scheduled purge cycle (configured by using the Customizer).
DataType	NUMBER	N	Y	Y	Foreign key reference to the BSADataTypes table.
DaysToRetain	INTEGER	N	N	N	Number of days to retain data for the current performance or alarm series.
HostID	\$ID\$	N	N	Y	Foreign key reference to the BSAHosts table.

3.23 BSATimeBand Table

The BSATimeBand table is used to store different time bands associated with an alarm event or performance series.

Table 3-22 *BSATimeBand Table*

Name	Type	P	M	F	Description
BandID	\$ID\$	Y	Y	N	Unique random number generated when the time band record was created.
AlphaDayOf Week	VARCHAR2(16)	N	N	N	Time band day of week represented by the day name (Monday, Tuesday Wednesday, etc.).
AlphaMonth	VARCHAR2(16)	N	N	N	Time band month of year represented by month name (January, February, etc.).
BandStart	\$DATETIME\$	N	Y	N	Date and time when the time band started.
DayofWeek	NUMBER	N	N	N	Numeric day of week associated with the current time band.
DayofYear	NUMBER	N	N	N	Numeric day of year associated with the current time band.
DayValue	NUMBER	N	N	N	The numeric day value for the current time band.
Granularity	INTEGER	N	N	N	Time band granularity. Currently not used.
HourValue	NUMBER	N	N	N	The hour value for the current time band.
MonthValue	NUMBER	N	N	N	The numeric month value for the current time band.
IsWeekDay	\$BOOLEAN\$	N	N	N	Boolean flag which indicates if this time band falls on a weekday.
WorkShift	NUMBER	N	N	N	The shift identifier associated with the current time band.
YearValue	NUMBER	N	N	N	The year value for the current time band.

3.24 BSAVersionWarehouse Table

The BSAVersionWarehouse table is an internal table used to manage the Service Warehouse schema.

Table 3-23 BSAVersionWarehouse Table

Name	Type	P	M	F	Description
Version	VARCHAR2(64)	N	Y	N	The Warehouse schema version number.
HostName	VARCHAR2(64)	Y	Y	N	Name of host that created the current Warehouse schema. This field is checked at Operations Center server startup to determine if the current Operations Center server created the schema. If the schema was created by a different host, an error is logged.
WhenCreated	\$TIMEPOINT\$	N	Y	N	The date/time when the Warehouse schema was created.

