Oracle® Business Intelligence Applications

Security Guide

Release 7.9.6.3

E19042-01

April 2011

ORACLE®

Oracle Business Intelligence Applications Security Guide, Release 7.9.6.3

E19042-01

Contents

Index

Preface

Oracle Business Intelligence Applications are comprehensive prebuilt solutions that deliver pervasive intelligence across an organization, empowering users at all levels — from front line operational users to senior management — with the key information they need to maximize effectiveness. Intuitive and role-based, these solutions transform and integrate data from a range of enterprise sources, including Siebel, Oracle, PeopleSoft, JD Edwards, and corporate data warehouses — into actionable insight that enables more effective actions, decisions, and processes.

Oracle BI Applications are built on Oracle Business Intelligence Suite Enterprise Edition, a comprehensive next-generation BI and analytics platform.

Oracle BI Applications includes the following:

- Oracle Contact Center Telephony Analytics
- Oracle Financial Analytics
- Oracle Human Resources Analytics
- Oracle Loyalty Analytics
- Oracle Marketing Analytics
- Oracle Pharma Marketing Analytics
- Oracle Pharma Sales Analytics
- Oracle Price Analytics
- Oracle Procurement and Spend Analytics
- Oracle Project Analytics
- Oracle Sales Analytics
- Oracle Service Analytics
- Oracle Supply Chain and Order Management Analytics

Oracle Business Intelligence Applications Security Guide contains information about the security features in Oracle BI Applications Release 7.9.6.3.

Oracle recommends reading the *Oracle Business Intelligence Applications Release Notes* before installing, using, or upgrading Oracle BI Applications. The *Oracle Business Intelligence Applications Release Notes* are available on the Oracle Business Intelligence Applications CD-ROM, or on the Oracle Technology Network at:

`http://www.oracle.com/technology/documentation/bi_apps.html`

Audience

This document is intended for BI managers and implementors of Oracle BI Applications.

> **Note:** Some tasks described in this guide might require you to retrieve information from your source system (for example, Oracle E-Business Suite). Such information is unique to your source system and implementation. To retrieve such information successfully, you need to have a good technical understanding of your source system. If you need assistance in obtaining information from your source system, you should consult with someone in your organization who possesses this knowledge, or consult the Oracle Support Services team for your source system.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/us/corporate/accessibility/index.html.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/support/contact.html or visit http://www.oracle.com/accessibility/support.html if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle BI Applications Release 7.9.6.3 documentation set (available at http://www.oracle.com/technology/documentation/bi_apps.html):

- *Oracle Business Intelligence Applications Release Notes*
- *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*
- *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*
- *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users*
- *Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users*
- *Oracle Business Intelligence Applications Naming Conventions and Domain Values Guide*
- *Oracle Business Analytics Warehouse Data Model Reference*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

What's New in This Release

This section lists security changes in Oracle Business Intelligence Applications Release 7.9.6.3.

1.1 What's New in Oracle Business Intelligence Applications Security for Release 7.9.6.3

Note the following changes to Oracle Business Intelligence Applications security in this release:

- RPD files now have RPD-specific passwords that are used to encrypt the contents. The RPD password is stored in an external credential store when you publish an RPD in Fusion Middleware Control, so that the Oracle BI Server can retrieve the password to load the RPD.
- Groups no longer exist in the RPD as objects. Instead, data access security is implemented based on the application roles to which a user belongs.
- Application roles are managed in an external policy store. Application role objects exist in the RPD, but these objects are pointers (references) to the externally managed roles.
- Users are managed in an external authentication provider and are no longer managed in the RPD. User objects exist in the RPD, but these objects are pointers (references) to the externally managed users.
- In this release, any named user can be granted administrative permissions if desired, unlike previous releases, in which there was a single user with administrative permissions who was named Administrator.

2

Integrating Security for Oracle BI Applications

This section describes the security features in Oracle Business Intelligence Applications. It contains the following main topics:

2.1 About Security in Oracle BI Applications

This section contains the following topics:

2.1.1 About Security Integration Between Oracle Business Enterprise Edition and Oracle BI Applications

Oracle BI Applications integrates tightly with Oracle Business Intelligence Enterprise Edition, as well as the security model of the operational source system, to allow the right content to be shown to the right user.

You should be thoroughly familiar with the security features of Oracle Business Intelligence Enterprise Edition before you begin working with Oracle BI Applications.

Security settings for Oracle Business Intelligence Enterprise Edition are made in the following Oracle Business Intelligence components:

- Oracle BI Administration Tool

  You can use the Oracle BI Administration Tool to perform tasks such as setting permissions for business models, tables, columns, and subject areas; specifying filters to limit data accessibility; and setting authentication options. For detailed information, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

- Oracle BI Presentation Services Administration

  You can use Oracle BI Presentation Services Administration to perform tasks such as setting permissions to Presentation Catalog objects, including dashboards and dashboard pages. For detailed information, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

- Oracle Enterprise Manager Fusion Middleware Control

  You can use Fusion Middleware Control to manage the policy store, application roles, and permissions for determining functional access. For detailed information, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

- Oracle WebLogic Server Administration Console

  You can use the Administration Console to manage users and groups in the embedded Oracle WebLogic Server LDAP. You can also use the Administration Console to manage security realms, and to configure alternative authentication providers. For detailed information, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

2.1.1.1 About User GUIDs in Oracle Business Intelligence Enterprise Edition

In Oracle Business Intelligence Enterprise Edition, users are recognized by their global unique identifiers (GUIDs), not by their names. GUIDs are identifiers that are completely unique for a given user. Using GUIDs to identify users provides a higher level of security because it ensures that data and metadata is uniquely secured for a specific user, independent of the user name.

Oracle recommends that you follow these two best practices to ensure that GUIDs are consistently applied in each phase of the development to production lifecycle:

- Ensure that a fan-out replica of the identity store is used between development, test, and production systems, so that user GUIDs are consistent and identical across the complete development to production lifecycle.
- Wherever possible, secure access to data and metadata using application roles rather than individual users.

Note that in cases where Oracle Business Intelligence Enterprise Edition test servers are configured against a test LDAP, and the production servers are configured against the corporate LDAP, but the test LDAP is *not* a fan-out copy of the corporate LDAP directory, a refresh of the LDAP GUIDs is needed for the system to function correctly. See "Regenerating User GUIDs" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition* for more information.

2.1.2 About Oracle BI Applications Security Levels

Security in Oracle BI Applications can be classified broadly into the following three levels:

- **User-level security (authentication of users).** User-level security refers to authentication and confirmation of the identity of a user based on the credentials provided. For more information, see Section 2.4, "User-Level Security in Oracle BI Applications."
- **Object-level security.** Object-level security controls the visibility to business logical objects based on a user's role. You can set up object-level security for metadata repository objects, such as business models and subject areas, and for Web objects, such as dashboards and dashboard pages, which are defined in the Presentation Catalog. For more information, see Section 2.3, "Object-Level Security in Oracle BI Applications."
- **Data-level security.** Data-level security controls the visibility of data (content rendered in subject areas, dashboards, Oracle BI Answers, and so on) based on the user's association to data in the transactional system. For more information, see Section 2.2, "Data-Level Security in Oracle BI Applications."

2.1.3 Using Application Roles in Oracle BI Applications

Object-level and data-level security are implemented in Oracle BI Applications using application roles. Application roles define a set of permissions granted to a user or group. Application roles are defined in the policy store using Oracle Enterprise Manager Fusion Middleware Control, and are composed of groups and users defined in LDAP. Application roles are typically related to either data or objects. For example, the Oracle BI Applications repository (OracleBIAnalyticsApps.rpd) uses the following application roles:

- The *HR Org-based Security* application role is used to control access to human resources data at the data security level.
- The *Human Resources Analyst* application role is used to control Presentation layer object visibility for the Human Resources Analyst role at the object security level.

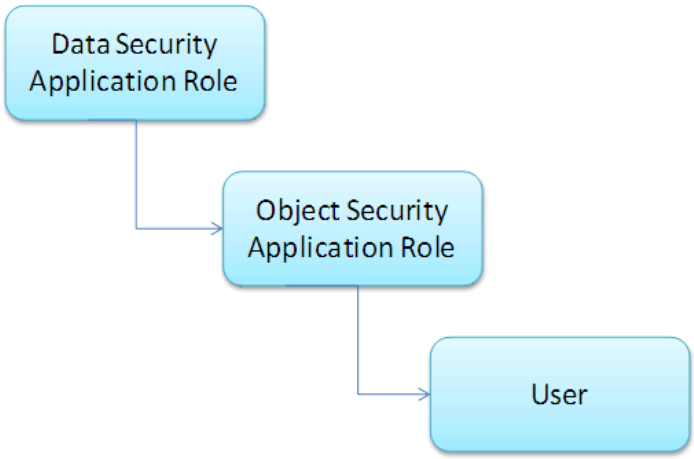
To view application roles in the RPD, select **Manage**, then select **Identity** in the Oracle BI Administration Tool. Application roles are visible in the Identity Manager dialog in online mode. In offline mode, only application roles that have had permissions, filters, or query limits set for them appear. For this reason, it is recommended that when you work with data access security in the Oracle BI Applications repository, you use the Administration Tool in online mode.

For detailed information about setting up and managing application roles, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*. For information about using the Oracle BI Administration Tool Identity Manager to apply data access security in the Oracle BI repository, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

The standard hierarchical structure of application roles and users in Oracle BI Applications is typically the following: data security application role, then object security application role, then user. It is a best practice to use this structure when setting up security.

Figure 2–1 shows the application role hierarchy in Oracle BI Applications.

Figure 2–1 Application Role Hierarchy in Oracle BI Applications



Use one of the following methods to set up application roles:

- Create application roles in the policy store with the same names as existing responsibilities or groups in the source applications. Then, add the new application roles as members of Oracle BI-specific application roles groups, and the users will inherit this membership based on their own responsibilities or roles in the OLTP application.
- Add new Oracle BI-specific responsibilities (Oracle EBS and Siebel CRM Applications) or roles (PeopleSoft Enterprise applications) in the source applications, making sure their names match the object security application roles in Oracle BI Applications, and assign OLTP users to these new groups. The users will then inherit the application role membership in the same way as described in the preceding method.

For information about integrating user and object security with JD Edwards, see Section 2.9, "About Security Integration with Oracle's JD Edwards EnterpriseOne or JD Edwards World."

2.1.4 Checking Oracle BI Applications User Responsibilities

An administrator can check a user's responsibility in the following ways:

- In the Siebel or Oracle EBS operational applications, go to the Responsibilities view.
- In PeopleSoft applications, go to the Roles view to check a user's roles.
- In JD Edwards EnterpriseOne applications, go to the User Profiles application (P0092) to check a user's roles.
- In JD Edwards World, go to User Information Revisions (P0092) to check a user's roles.
- Individual users can view the list of application roles to which they are assigned. In the Oracle BI application, click **Signed In As** ***username*** and select **My Account**. Then, click the Roles and Catalog Groups tab to view the application roles. In Presentation Services, application roles are used to control the ability to perform actions (privileges) within Presentation Services.

For more information, refer to the system administrator for your source system.

2.1.5 About Managing Presentation Services Catalog Privileges in Oracle Business Intelligence

When you add a new catalog privilege to an application role in Oracle BI Presentation Services, the change is not immediately reflected in the Oracle Business Intelligence environment. In order to register the catalog privilege, both the administrator and the user must perform the following tasks:

1. The Oracle BI administrator must reload the Oracle BI Server metadata through Oracle BI Presentation Services. To reload the metadata, in Oracle Business Intelligence Answers, select **Administration**, and then click **Reload Files and Metadata**.

   For more information on managing Presentation Services catalog privileges, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

2. Users belonging to that application role must log out from the Oracle BI application (or from Siebel or Oracle EBS operational application if the user is looking at Oracle BI dashboards using an embedded application) and then log in again.

2.2 Data-Level Security in Oracle BI Applications

This section describes the data-level security features in Oracle BI Applications. It contains the following topics:

- Section 2.2.1, "Overview of Data-Level Security in Oracle BI Applications"
- Section 2.2.2, "Implementing Data-Level Security in the Oracle BI Repository"
- Section 2.2.3, "Initialization Blocks Used for Data-Level Security in Oracle BI Applications"
- Section 2.2.4, "Data-Level Security Application Roles in Oracle BI Applications"
- Section 2.2.5, "About Data-Level Security Design in Oracle BI Applications"

2.2.1 Overview of Data-Level Security in Oracle BI Applications

Data-level security defines what a user in an OLTP application can access inside a report. The same report, when run by two different users, can bring up different data. This is similar to how the My Opportunities view in an operational application displays different data for different users. However, the structure of the report is the same for all users, unless a user does not have access to the report subject area, in which case the report displays an error.

Table 2–1 shows the data security policies that are supported in Oracle BI Applications. During installation and configuration, you must make sure the correct application roles and initialization blocks are set up for your environment.

For more information about the use of initialization blocks in Oracle Business Intelligence, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition* and *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

Table 2–1 Summary of Supported Data Security Policies

Application Role	Oracle EBS	PeopleSoft: Financials	PeopleSoft: HCM	PeopleSoft: Projects	PeopleSoft: Procurement and Spend	Siebel
Operating Unit Org-based Security	Available since 7.9.3	Available since 7.9.3	Not Available	Available since 7.9.6	Available since 7.9.6	Not Available
Company Org-based Security	Available in 7.9.3 and obsolete in 7.9.4	Available since 7.9.3	Available since 7.9.3	Not Available	Not Available	Not Available
Business Group Org-based Security	Available since 7.9.3	Not Available	Not Available	Not Available	Not Available	Not Available
HR Org-based Security	Available since 7.9.3	Not Available	Available since 7.9.3 and enhanced in 7.9.6 to support PeopleSoft department security	Not Available	Not Available	Not Available
Inventory Org-based Security	Available since 7.9.3	Not Available	Not Available	Not Available	Not Available	Not Available
Payable Org-based Security	Not Available	Available since 7.9.3	Not Available	Not Available	N/A	Not Available
Receivables Org-based Security	Not Available	Available since 7.9.3	Not Available	Not Available	N/A	Not Available
SET ID-based Security	Not Available	Available since 7.9.3	Available since 7.9.3	Available since 7.9.6	Available since 7.9.6	Not Available
Primary Employee/Position Hierarchy-based Security	Available since 7.9.4 for HRMS	Not Available	Available since 7.9.3	Not Available	N/A	Available since 7.5
Ledger-based Security	Available since 7.9.4	Available since 7.9.4	Not Available	Not Available	Available since 7.9.6	Not Available

2.2.2 Implementing Data-Level Security in the Oracle BI Repository

Data-level security in Oracle BI Applications is implemented in three major steps, as described below. For instructions on performing these steps, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition.*

1. Set up initialization blocks that obtain specific security-related information when a user logs in, for example, the user's hierarchy level in the organization hierarchy, or the user's responsibilities. Initialization blocks obtain DimensionIds for each user session in order to restrict row-level access to factual or dimensional data.

   For a description of the preconfigured initialization blocks, see Section 2.2.3, "Initialization Blocks Used for Data-Level Security in Oracle BI Applications."

2. Set up the joins to the appropriate security tables in the metadata physical and logical layers.

   For detailed information about this security feature, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

3. Set up the data filters for each application role on each logical table that needs to be secured.

For detailed information about this security feature, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

2.2.3 Initialization Blocks Used for Data-Level Security in Oracle BI Applications

For more information about setting up and managing initialization blocks, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

In the Oracle BI Repository, the initialization blocks are set up for obtaining a given user's primary position, primary organization, and the owner ID, as described below:

- **Authorization**

  This initialization block is used to associate users with all application roles to which they belong. It obtains a user's responsibilities or roles from the source OLTP application, matches them with Oracle BI Applications application roles, and determines the user's applicable object security during the session. This initialization block populates a variable set called GROUP.

- **Business Groups**

  This initialization block is used to retrieve the business groups from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called BUSINESS_GROUP, which is used to drive security permissions for business group org-based security.

- **Companies**

  This initialization block is used to retrieve the companies from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called COMPANY, which is used to drive security permissions for company org-based security. COMPANY is mapped to the PeopleSoft business unit.

- **HR Organizations**

  This initialization block is used to retrieve the HR organizations from the OLTP application to which the corresponding login user has access. This initialization block populates a variable set called HR_ORG, which is used to drive security permissions for HR analysts.

- **Inventory Organizations**

  This initialization block is used to retrieve the inventory organizations from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called INV_ORG, which is used to drive security permissions for inventory org-based security.

- **Ledgers**

  This initialization block is used to retrieve the ledgers from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called LEDGER, which is used to drive security permissions for ledger-based security.

- **Operating Unit Organizations**

  This initialization block is used to retrieve the operating unit organizations from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called OU_ORG, which is used to drive security permissions for operating unit org-based security.

- **Orgs for Org-Based Security**

  This initialization block is used to retrieve the organizations reporting to the current user's business unit. This initialization block populates a variable set called ORGANIZATION, which is used to drive primary org-based security.

- **Payable Organizations**

  This initialization block is used to retrieve the payable organizations from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called PAYABLE_ORG, which is used to drive security permissions for payable org-based security.

- **Primary Owner ID**

  This initialization block obtains the owner ID for the given user. It obtains this information from the Siebel OLTP and populates the PR_OWNER_ID variable.

- **Payables Organizations**

  This initialization block is used to retrieve the payables organizations from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called PAYABLE_ORG, which is used to drive security permissions for payables org-based security.

- **SetID**

  This initialization block is used to retrieve the set IDs from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called SET_ID, which is used to drive security permissions for Set ID-based security.

- **User Hierarchy Level**

  This initialization block obtains the fixed hierarchy level of the given user, based on the user's login, from W_POSITION_DH. It populates the variable HIER_LEVEL. The SQL used by the block is run against the data warehouse. Therefore, it reflects the hierarchy level at the time of the last ETL run that populated this table (W_POSITION_DH).

- **User HR Organizations**

  This initialization block is used to retrieve the current HR organization from OLTP application to which the current user belongs. This initialization block populates a variable called USER_HR_ORG.

2.2.4 Data-Level Security Application Roles in Oracle BI Applications

Table 2–2 describes the application roles used in Oracle BI Applications and the application to which they apply. Some selected application roles share the same name as responsibilities for Siebel CRM and Oracle EBS applications and roles for PeopleSoft applications. A user who has any of these responsibilities or roles in the source application will be a member of the corresponding application role automatically upon logging in to the Oracle BI application. Other application roles based on similar objects in the source application can be added to the policy store, and then added to these data-level application roles, if you need the corresponding data filters to apply to any additional group of users. Table 2–2 shows the application roles that are supported in Oracle BI Applications.

Table 2–2 Data-Level Security Application Roles in Oracle BI Applications

Application Role Name	Supported Source Application	Description	Associated Initialization Block Name
Business Group Org-Based Security	Oracle EBS, PeopleSoft HR	A business group is the highest level in the organization structure and is usually used to represent the entire enterprise or a major division. A business group can have several sets of books.	Business Groups
Company Org-Based Security	PeopleSoft HR and Financials	This application role filters data based on the GL or HR business units associated to the user that is logged in. The business unit is the highest level key structure in PeopleSoft.	Companies
HR Org-Based Security	PeopleSoft HR	This application role filters data based on the HR organizations that the user is authorized to see. This security works in conjunction with HR Organization hierarchy to restrict users to access organizations that they are authorized to view and the organizations that report to them.	HR Organizations
Human Resource Personnel Data Security	Oracle EBS, PeopleSoft HR	This application role gives HR staff access to all organizations that they are authorized to see (except for their own organization) and to the supervisors they are authorized to see based on the Supervisor (W_POSITION_DH) security.	HR Organizations User HR Organizations
Inventory Org-Based Security	Oracle EBS	An inventory organization tracks inventory transactions and balances, and/or manufactures or distributes products or components. This application role filters data based on the inventory orgs associated to the user that is logged in.	Inventory Organizations
Ledger-Based Security	Oracle EBS, PeopleSoft Financials	A ledger is essentially a reporting organization that uses a common chart of accounts, functional currency, fiscal calendar, and accounting method. This application role filters data based on the ledgers associated to the user that is logged in.	Ledgers
Operating Unit Org-Based Security	Oracle EBS, PeopleSoft Financials, Siebel CRM	This application role filters data based on the organizations associated to the user that is logged in.	Operating Unit Organizations
Payables Org-Based Security	PeopleSoft Financials	This application role filters data based on the payables business units associated to the user that is logged in. The business unit is the highest level key structure in PeopleSoft.	Payables Organizations

Table 2–2 (Cont.) Data-Level Security Application Roles in Oracle BI Applications

Application Role Name	Supported Source Application	Description	Associated Initialization Block Name
Primary Employee/Position Hierarchy-Based Security	Oracle EBS, PeopleSoft HR, PeopleSoft: Procurement and Spend, PeopleSoft Financials, Siebel CRM	This application role allows managers to view employee data in their supervisory hierarchy, including their direct reports and those reporting up the chain of command. **Note:** Primary Employee/Position Hierarchy-Based security is not available for Siebel Service Analytics. The security available for Siebel Service Analytics is visibility granted to the primary owner organization.	User Hierarchy Level
Primary Owner-Based Security	Siebel CRM	This application role filters data based on the user that is logged.	Primary Owner ID
Receivables Org-Based Security	PeopleSoft Financials, Siebel CRM	This application role filters data based on the receivables business units associated to the user that is logged in. The business unit is the highest level key structure in PeopleSoft.	Receivables Organizations
SET ID-Based Security	PeopleSoft Financials, Oracle EBS	This application role filters data based on the Set IDs associated to the user that is logged in.	Set ID

2.2.5 About Data-Level Security Design in Oracle BI Applications

As discussed in the preceding sections, Oracle BI Applications maintains data-level security application roles that are assigned dynamically to every user at the session level. Each application role has a set of filters associated with it that determines the data each user is allowed to see. A user is assigned an application role through the Authorization initialization block, as discussed in Section 2.2.3, "Initialization Blocks Used for Data-Level Security in Oracle BI Applications."

The data security design has the following features:

- **Drill down.** The user can drill down on a particular position in the position hierarchy to slice the data by the next position level in the hierarchy. For example, if the initial report is defined as:

  ```
  select Top Level Position, Revenue from RevenueStar
  ```

  then by drilling down on a value of MyPosition in the TopLevelPosition hierarchy, the report will become:

  ```
  Select Level8 Position, Revenue, where TopLevelPosition = 'MyPosition'
  ```

- **Personalized reports.** Users at different levels of the Position hierarchy can use the same Position-based reports but with each user seeing the data corresponding to his or her level. In such reports, Position is a dynamic column.

  For example, if a report is defined as:

  ```
  select Position, Revenue from RevenueStar
  ```

  the logical query for the user at the top level of the hierarchy will be:

  ```
  select Top Level Position, Revenue from RevenueStar
  ```

The logical query for the user at the next level of the hierarchy will be:

```
select Level8 Position, Revenue from RevenueStar
```

- **CURRENT Position hierarchy columns.** Position hierarchy columns with the prefix CURRENT contain the Current Position hierarchy at any point of time. This feature allows users to see the same data associated with the employee holding the Current Employee position at the time the report runs. This type of Analysis is called As Is.
- **Additional Position hierarchy columns.** The columns EMP_LOGIN and EMPLOYEE_FULL_NAME are used at every level of the Position hierarchy to store additional information about an employee holding a particular position. In the Logical layer, the Employee path and Position path are two drill down paths under the Position hierarchy that allow the user to drill down on a position to see all positions under it. It also allows an employee to see all the employees reporting to him or her.

2.3 Object-Level Security in Oracle BI Applications

This section describes the object-level security features in Oracle BI Applications. It contains the following topics:

2.3.1 Metadata Object-Level Security in the RPD

Application roles control access to metadata objects, such as subject areas, tables and columns. For example, users in a particular department can view only the subject areas that belong to their department.

Metadata object security is configured in the Oracle BI Repository, using the Oracle BI Administration Tool. The Everyone application role is denied access to each of the subject areas. Each subject area is configured to give explicit read access to selected related responsibilities. This access can be extended to tables and columns.

> **Note:** By default in Oracle BI Applications, only permissions at the subject area level have been configured.

Note: The Siebel Communications and Financial Analytics industry applications have tables and columns that are industry-specific, and, therefore, hidden from other application roles.

Oracle Business Intelligence supports hierarchies within application roles. In the policy store, there are certain application roles that are parent application roles, which define the behavior of all the child application roles. Inheritance is used to let permissions ripple through to child application roles. The parent application roles and their purpose are shown in Table 2–3.

Table 2–3 Parent Application Roles

Parent Application Roles	Permissions Inherited By
Finance	All application roles for Financial applications

Table 2–3 (Cont.) Parent Application Roles

Parent Application Roles	Permissions Inherited By
Insurance	All application roles for Insurance applications
CM General	All application roles for Communications applications
Consumer Sector	All application roles for Consumer Sector
Pharma	All application roles for Life Sciences/Pharmaceuticals applications
Channel Managers	All application roles for Channel applications
Partner Managers	All application roles for Partner applications

2.3.2 Metadata Object-Level Security in Presentation Services

Access to Oracle BI Presentation Services objects, such as dashboards, pages, reports, and Web folders, are controlled using application roles. For detailed information about managing object-level security in Presentation Services, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

2.4 User-Level Security in Oracle BI Applications

User security concerns the authentication and confirmation of the identity of the user based on the credentials provided, such as username and password. By default, user-level security is set up in the embedded Oracle WebLogic Server LDAP server and policy store in Oracle Business Intelligence Enterprise Edition. For more information, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

2.5 Extending Security in Oracle BI Applications

You can extend the preconfigured Oracle BI Applications security model to match your operational source system. When you extend Oracle BI Applications, you need to ensure that your customizations and any new objects are valid and functional.

The general process for extending data-level security for repository objects is as follows:

1. Extend the physical table by adding the attribute by which the dimension or fact needs to be secured. (This step results in a change to the data model.)
2. Populate the relevant attribute value for each row in the fact or dimension table. (This step results in a change to the ETL mapping.)
3. Use the Oracle BI Administration Tool to create an initialization block to fetch the attribute values and populate them into a session variable when each user logs into Oracle BI Applications. You can create a target session variable for the initialization block if the initialization block is not a row-wise initialization block. (This step results in a change to the Oracle BI repository.) For instructions, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.
4. Use Fusion Middleware Control to create an application role in the policy store. Then, restart the Oracle BI Server. For instructions, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.
5. Use the Oracle BI Administration Tool in online mode to set up data filters based on the new role for each of the fact and dimension tables that need to be secured

by the attribute you added in Step 1. (This step results in a change to the Oracle BI Repository.) For instructions, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.

6. Use the Oracle BI Administration Tool in online mode to restrict object access based on the application role you created in Step 4. (This step results in a change to the Oracle BI Repository.) For instructions, see *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*.
7. Use Presentation Services administration to set up Presentation Services catalog privileges based on the application role you created in step 4. For instructions, see *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

Note: You can also leverage the existing Oracle BI Applications security objects when extending data-level security. To do this, copy existing security objects for secured dimensions, such as initialization blocks and application roles, and then modify them to apply to the additional dimensions.

For more information on working with security objects like application roles and initialization blocks, see the following resources:

- "Creating and Managing Application Roles and Application Policies Using Fusion Middleware Control" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*
- "Working with Initialization Blocks" in *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition*

2.6 Integrating Data Security for Oracle EBS

This section explains how security in Oracle BI Applications is deployed with Oracle EBS. Read this section if you want to understand how the default security settings are configured so that you can change the way security is implemented if required. This section contains the following topics:

2.6.1 Oracle BI Applications Authorization for Oracle EBS

The authorization process of Oracle BI Applications fetches a user's responsibilities from source Oracle EBS applications, matches them with all Oracle BI Applications application roles, and determines the user's applicable object security during a user's session. The initialization block Authorization is used to fetch roles and assign the

result set to a special session variable called GROUP. The initialization block SQL is the following:

```
SELECT DISTINCT 'GROUP', RESPONSIBILITY_NAME FROM
FND_USER,FND_USER_RESP_GROUPS, FND_RESPONSIBILITY_VL
WHERE
FND_USER.user_id=FND_USER_RESP_GROUPS.user_id
AND FND_USER_RESP_GROUPS.RESPONSIBILITY_ID = FND_RESPONSIBILITY_VL.RESPONSIBILITY_
ID
AND FND_USER_RESP_GROUPS.RESPONSIBILITY_APPLICATION_ID = FND_RESPONSIBILITY_
VL.APPLICATION_ID AND
FND_USER_RESP_GROUPS.START_DATE < SYSDATE AND
(CASE WHEN FND_USER_RESP_GROUPS.END_DATE IS NULL THEN SYSDATE ELSE TO_DATE(FND_
USER_RESP_GROUPS.end_Date) END) >= SYSDATE
AND FND_USER.user_id = (SELECT USER_ID FROM FND_USER WHERE UPPER(USER_NAME =
UPPER('VALUEOF(NQ_SESSION.USER)')))
```

2.6.2 Operating Unit-Based Security for Oracle EBS

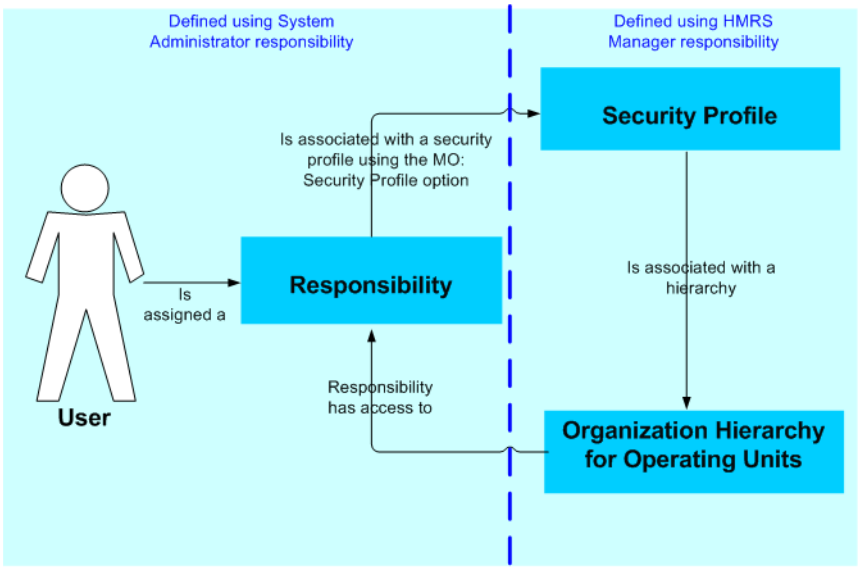
This section contains the following topics:

- Section 2.6.2.1, "About Operating Unit-Based Security for Oracle EBS"
- Section 2.6.2.2, "Implementation Steps for Operating Unit-Based Security for Oracle EBS"

2.6.2.1 About Operating Unit-Based Security for Oracle EBS

Operating units are secured by attaching a security profile to a user ID or a responsibility. In turn, a security profile is associated with an organization hierarchy, which also has access to the user ID or responsibility (see Figure 2–2). The user ID or responsibility is defined using the System Administrator responsibility. The security profile and organization hierarchy are defined using the HRMS Manager responsibility.

Figure 2–2 Operating Unit-Based Security for Oracle EBS



Operating Unit assignment is decided by looking at the profiles set at the following levels, with the order of precedence indicated:

1. User
2. Responsibility
3. Application
4. Site

In other words, if a value is set in the profile at the user level and at the site level, the value set at the user level takes precedence.

2.6.2.2 Implementation Steps for Operating Unit-Based Security for Oracle EBS

The sequence for operating unit-based security for Oracle EBS is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

   ```
   USER (System variable)
   ```

2. The 'EBS Single Sign-on Integration' session variable is initialized in the 'EBS Single Sign-on Integration' initialization block:

   ```
   EBS_SSO_INTEGRATION_MODE
   ```

   This session can be initialized with two possible values, 'Integrated' or 'Not Integrated', to indicate whether or not Oracle BI Applications is integrated with EBS SSO.

3. The 'EBS Security Context' initialization block then populates these session variables:

   ```
   OLTP_EBS_RESP_ID
   ```

   The session variable is initialized with the responsibility of the user's session in Oracle EBS if Oracle BI Applications is integrated with EBS; otherwise, it is defaulted to a random value, which will be ignored.

   ```
   OLTP_EBS_RESP_APPL_ID
   ```

   The session variable is initialized with the responsibility application of the user session in EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

4. The Oracle BI Server will get the operating unit corresponding to the USER from FND_USER_RESP_GROUPS. The following session variable is set automatically:

   ```
   OU_ORG (Row-wise variable)
   ```

   The initialization block 'Operating Unit Organizations', which sets the value for this variable, is shown below.

   Initialization block -- 'Operating Unit Organizations'

   The initialization block 'Operating Unit Organizations' sets the value for variable OU_ORG using the following SQL:

   ```
   SELECT DISTINCT 'OU_ORG', TO_CHAR(PER_ORGANIZATION_
   LIST.ORGANIZATION_ID)

   FROM PER_ORGANIZATION_LIST,

   (SELECT FND_PROFILE.VALUE_SPECIFIC('XLA_MO_SECURITY_PROFILE_
   LEVEL', USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_
   APPLICATION_ID) PROFILE_ID
   ```

```
FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_
APPLICATION_ID

FROM FND_USER_RESP_GROUPS

WHERE START_DATE < SYSDATE

AND (CASE WHEN END_DATE IS NULL THEN SYSDATE ELSE TO_
DATE(END_DATE) END) >= SYSDATE

AND USER_ID = (SELECT USER_ID FROM FND_USER WHERE UPPER(USER_
NAME = UPPER('VALUEOF(NQ_SESSION.USER)'))

AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_
SSO_INTEGRATION_MODE) = 'Integrated' THEN

VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID
END)

AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN VALUEOF(NQ_
SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated' THEN
VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END)))

WHERE PER_ORGANIZATION_LIST.SECURITY_PROFILE_ID = PROFILE_ID

UNION

SELECT DISTINCT 'OU_ORG', FND_PROFILE.VALUE_SPECIFIC('ORG_
ID', USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_
ID) ORGANIZATION_ID

FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_
APPLICATION_ID

FROM FND_USER_RESP_GROUPS

WHERE START_DATE < SYSDATE

AND (CASE WHEN END_DATE IS NULL THEN SYSDATE ELSE TO_
DATE(END_DATE) END) >= SYSDATE

AND USER_ID = (SELECT USER_ID FROM FND_USER WHERE UPPER(USER_
NAME) = UPPER('VALUEOF(NQ_SESSION.USER)'))

AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_
SSO_INTEGRATION_MODE) = 'Integrated' THEN VALUEOF(NQ_
SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END)

AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN VALUEOF(NQ_
SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated' THEN
VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END))
```

2.6.3 Inventory Org-Based Security for Oracle EBS

This section contains the following topics:

- Section 2.6.3.1, "About Inventory Org-Based Security for Oracle EBS"
- Section 2.6.3.2, "Implementation Steps for Inventory Org-Based Security for Oracle EBS"

2.6.3.1 About Inventory Org-Based Security for Oracle EBS

With inventory org-based security, the organization that a user belongs to determines which rows of data they can access. Inventory org-based security is applied based on the current logged-in responsibility rather than the current user. With Oracle EBS sources, an inventory organization can be associated with multiple responsibilities.

2.6.3.2 Implementation Steps for Inventory Org-Based Security for Oracle EBS

The sequence for inventory org-based security for Oracle EBS is described below:

1. When a user logs in to Oracle BI Applications, the following session variable is set automatically.

   ```
   USER (System variable)
   ```

2. The 'EBS Single Sign-on Integration' session variable is initialized in the 'EBS Single Sign-on Integration' initialization block:

   ```
   EBS_SSO_INTEGRATION_MODE
   ```

   This session can be initialized with two possible values, 'Integrated' or 'Not Integrated', to indicate whether Oracle BI Applications is integrated with EBS SSO or not.

3. The 'EBS Security Context' initialization block then populates these session variables:

   ```
   OLTP_EBS_RESP_ID
   ```

   The session variable is initialized with the responsibility of the user session in Oracle EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

   ```
   OLTP_EBS_RESP_APPL_ID
   ```

   The session variable is initialized with the responsibility application of the user session in EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

4. The Oracle BI Server will get the inventory org corresponding to the USER from FND_USER_RESP_GROUPS. The following session variable is set automatically:

   ```
   INV_ORG (Row-wise variable)
   ```

   The initialization block 'Inventory Organizations', which sets the value for this variable, is shown below.

   Initialization block -- 'Inventory Organizations'

   The initialization block 'Inventory Organizations' sets the value for variable INV_ ORG using the following SQL:

   ```
   SELECT DISTINCT 'INV_ORG', BIS_ORGANIZATIONS_V.ID

   FROM FND_USER_RESP_GROUPS, BIS_ORGANIZATIONS_V

   WHERE FND_USER_RESP_GROUPS.RESPONSIBILITY_ID = BIS_
   ORGANIZATIONS_V.RESPONSIBILITY_ID

   AND FND_USER_RESP_GROUPS.START_DATE < SYSDATE

   AND (CASE WHEN FND_USER_RESP_GROUPS.END_DATE IS NULL THEN
   SYSDATE ELSE
   ```

```
AND FND_USER_RESP_GROUPS.USER_ID = (SELECT USER_ID FROM FND_
USER WHERE UPPER(USER_NAME) = UPPER('VALUEOF(NQ_
SESSION.USER)'))

AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_
SSO_INTEGRATION_MODE) = 'Integrated' THEN

VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID
END)

AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN VALUEOF(NQ_
SESSION.EBS_SSO_INTEGRATION_MODE) =

'Integrated' THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID)
ELSE RESPONSIBILITY_APPLICATION_ID END)
```

2.6.4 Ledger-Based Security for Oracle EBS

Ledger-based security for Oracle EBS was introduced in Oracle BI Applications release 7.9.4. It replaces the company-based security to support the Oracle EBS GL set of books.

This section contains the following topics:

- Section 2.6.4.1, "About Ledger-Based Security for Oracle EBS"
- Section 2.6.4.2, "Implementation Steps for Ledger-Based Security for Oracle EBS"

2.6.4.1 About Ledger-Based Security for Oracle EBS

In Oracle EBS Release 11i, a set of books is essentially a reporting entity that defines the reporting context including a chart of accounts, a functional currency, and an accounting calendar. A set of books can be assigned to a user, a responsibility, or to the site as the default for all responsibilities. Each user is associated with a single set of books when they log in to the application under a given responsibility in Oracle Applications. Ledger-based security filters data based on the set of books associated with the user that is logged in.

In Oracle EBS Release 12, the set of books is replaced by the ledger. A ledger determines the currency, chart of accounts, accounting calendar, ledger processing options and subledger accounting method. The data access set assigned to the user's responsibility controls what ledgers the user can access. A user may be able to access multiple ledgers from a responsibility. Ledger-based security filters data based on the ledgers associated with the user that is logged in.

2.6.4.2 Implementation Steps for Ledger-Based Security for Oracle EBS

The sequence for ledger-based security for Oracle EBS is described below:

1. When a user logs in to Oracle Business Intelligence Enterprise Edition, the session variable below is set automatically.

   ```
   USER (System variable)
   ```

2. The 'EBS Single Sign-on Integration' session variable is initialized in the 'EBS Single Sign-on Integration' initialization block:

   ```
   EBS_SSO_INTEGRATION_MODE
   ```

   This session can be initialized with two possible values, 'Integrated' or 'Not Integrated', to indicate whether Oracle BI Applications is integrated with EBS SSO or not.

3. The 'EBS Security Context' initialization block then populates these session variables:

   OLTP_EBS_RESP_ID

   The session variable is initialized with the responsibility of the user session in Oracle EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

   OLTP_EBS_RESP_APPL_ID

   The session variable is initialized with the responsibility application of the user session in EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

4. Then this session variable would be initialized in another init block, “Ledgers”, which gets the ledgers (which is essentially the set of books in EBS) corresponding to the USER and OLTP_EBS_RESP_ID and OLTP_EBS_RESP_APPL_ID, via table FND_USER_RESP_GROUPS and procedure FND_PROFILE.

   Row-wise variable:

   LEDGER (Row-wise variable)

5. The Oracle BI server gets the set of books or ledgers corresponding to the USER and OLTP_EBS_RESP_ID from the OLTP. The 'Ledgers' initialization block then populates these session variables.

The Ledgers initialization block should be set according to the Oracle EBS release, as follows:

- If you are using EBS release 12 or after, the following SQL applies as the data source in the initialization block:

```
SELECT DISTINCT 'LEDGER', TO_CHAR(GAL.LEDGER_ID)

FROM GL_ACCESS_SET_LEDGERS GAL, (SELECT FND_PROFILE.VALUE_
SPECIFIC('GL_ACCESS_SET_ID',USER_ID, RESPONSIBILITY_ID,
RESPONSIBILITY_APPLICATION_ID) PROFILE_VALUE

FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_
APPLICATION_ID

FROM FND_USER_RESP_GROUPS

WHERE START_DATE < SYSDATE AND (CASE WHEN END_DATE IS NULL
THEN SYSDATE ELSE

TO_DATE(END_DATE) END) >= SYSDATE AND USER_ID = (CASE WHEN

'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' = 'Integrated'

THEN VALUEOF(NQ_SESSION.OLTP_EBS_USER_ID) ELSE (SELECT USER_
ID FROM FND_USER WHERE

USER_NAME = 'OPERATIONS') END) AND RESPONSIBILITY_ID = (CASE
WHEN

'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' = 'Integrated'

THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE
RESPONSIBILITY_ID END)

AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN

'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' = 'Integrated'
```

```
THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END)

))WHERE GAL.ACCESS_SET_ID = PROFILE_VALUE
```

- If you are using Oracle EBS 11i, the following SQL applies as the data source in the Ledgers initialization block:

```
SELECT DISTINCT 'LEDGER', FND_PROFILE.VALUE_SPECIFIC('GL_SET_
OF_BKS_ID', USER_ID,

RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID)

FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_
APPLICATION_ID FROM

FND_USER_RESP_GROUPS

WHERE START_DATE < SYSDATE

AND (CASE WHEN END_DATE IS NULL THEN SYSDATE ELSE TO_
DATE(END_DATE) END) >= SYSDATE

AND USER_ID IN (CASE WHEN VALUEOF(NQ_SESSION.EBS_SSO_
INTEGRATION_MODE) = 'Integrated'

THEN VALUEOF(NQ_SESSION.OLTP_EBS_USER_ID) ELSE (SELECT USER_
ID FROM FND_USER WHERE UPPER(USER_NAME) = UPPER('VALUEOF(NQ_
SESSION.USER)')) END)

AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_
SSO_INTEGRATION_MODE) = 'Integrated'

THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE
RESPONSIBILITY_ID END)

AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN

VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated'

THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END))
```

2.6.5 Business Group Org-Based Security for Oracle EBS

This section contains the following topics:

- Section 2.6.5.1, "About Business Group Org-Based Security for Oracle EBS"
- Section 2.6.5.2, "Implementation Steps for Business Group Org-Based Security for Oracle EBS"

2.6.5.1 About Business Group Org-Based Security for Oracle EBS

A business group is the highest level in the organization structure. It is usually used to represent the entire enterprise or a major division. A business group can have several sets of books.

2.6.5.2 Implementation Steps for Business Group Org-Based Security for Oracle EBS

The sequence for business group org-based security for Oracle EBS is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

```
USER (System variable)
```

2. The 'EBS Single Sign-on Integration' session variable is initialized in the 'EBS Single Sign-on Integration' initialization block:

   ```
   EBS_SSO_INTEGRATION_MODE
   ```

   This session can be initialized with two possible values, 'Integrated' or 'Not Integrated', to indicate whether Oracle BI Applications is integrated with EBS SSO or not.

3. The 'EBS Security Context' initialization block then populates these session variables:

   ```
   OLTP_EBS_RESP_ID
   ```

   The session variable is initialized with the responsibility of the user session in Oracle EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

   ```
   OLTP_EBS_RESP_APPL_ID
   ```

   The session variable is initialized with the responsibility application of the user session in EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

4. The Oracle BI Server will get the business group corresponding to the USER and OLTP_EBS_RESP_ID from FND_USER_RESP_GROUPS. The following session variable is set automatically:

   ```
   BUSINESS_GROUP (Row-wise variable)
   ```

   The initialization block 'Business Groups', which sets the value for this variable, is shown below.

   Initialization block -- 'Business Groups'

   The initialization block 'Business Groups' sets value for variable INV_ORG using the following SQL:

   ```
   SELECT DISTINCT 'BUSINESS_GROUP',

   TO_CHAR(FND_PROFILE.VALUE_SPECIFIC('PER_BUSINESS_GROUP_ID',
   USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID))

   FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_
   APPLICATION_ID FROM FND_USER_RESP_GROUPS WHERE START_DATE <
   SYSDATE AND (CASE WHEN END_DATE IS NULL THEN SYSDATE ELSE TO_
   DATE(END_DATE) END) >= SYSDATE AND USER_ID = (CASE WHEN
   'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' = 'Integrated'
   THEN VALUEOF(NQ_SESSION.OLTP_EBS_USER_ID) ELSE (SELECT USER_
   ID FROM FND_USER WHERE UPPER(USER_NAME) = UPPER('VALUEOF(NQ_
   SESSION.USER)')) END)

   AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_
   SSO_INTEGRATION_MODE) = 'Integrated' THEN VALUEOF(NQ_
   SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END)

   AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN VALUEOF(NQ_
   SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated' THEN
   VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
   RESPONSIBILITY_APPLICATION_ID END))
   ```

Note: The 'Business Group Org-Based Security' application role contains all the data access permission filters.

2.6.6 HR Org-Based Security for Oracle EBS

This section contains the following topics:

2.6.6.1 About HR Org-Based Security for Oracle EBS

HR org-based security for Oracle EBS supports the standard HRMS organization security defined in Oracle EBS Human Resources. Oracle EBS Human Resources restricts access by organization, position, and payroll based on the security policies defined in the security profile.

2.6.6.2 Implementation Steps for HR Org-Based Security for Oracle EBS

The sequence for HR org-based security for Oracle EBS is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

   ```
   USER (System variable)
   ```

2. The Oracle BI Server gets the HR organizations corresponding to the USER from the following tables:
   - FND_USER_RESP_GROUPS
   - FND_USER
   - PER_SECURITY_PROFILES
   - PER_SEC_PROFILE_ASSIGNMENTS
   - PER_PERSON_LIST

     Note: Before the PER_PERSON_LIST table can be used, you must ensure that you have run the Oracle EBS HRMS Security List Maintenance process.
   - PER_ALL_ASSIGNMENTS_F
3. The following session variable is set automatically:

   ```
   HR_ORG (Row-wise variable)
   ```

   The initialization block 'HR Organizations', which sets the value for this variable, is shown below.

   Initialization block -- 'HR Organizations'

   The initialization block 'HR Organizations' sets value for variable HR_ORG using the following SQL. The actual SQL query differs depending on whether Multiple Security Group (MSG) is set up or not.

   The following SQL should be used when MSG is not in place:

   ```
   SELECT
    DISTINCT 'HR_ORG'
   ,TO_CHAR(SEC_DET.ORGANIZATION_ID)
   FROM
   (
   SELECT
     'HR_ORG',
     ASG.ORGANIZATION_ID
   FROM
    FND_USER_RESP_GROUPS URP
   ```

```
,FND_USER USR
,PER_SECURITY_PROFILES PSEC
,PER_PERSON_LIST PER
,PER_ALL_ASSIGNMENTS_F ASG
WHERE
  URP.START_DATE < TRUNC(SYSDATE)
AND (CASE WHEN URP.END_DATE IS NULL THEN TRUNC(SYSDATE) ELSE TO_DATE(URP.END_
DATE) END) >= TRUNC(SYSDATE)
AND USR.USER_NAME = UPPER('VALUEOF(NQ_SESSION.USER)')
AND USR.USER_ID = URP.USER_ID
AND TRUNC(SYSDATE)
  BETWEEN URP.START_DATE AND NVL(URP.END_DATE, HR_GENERAL.END_OF_TIME)
AND PSEC.SECURITY_PROFILE_ID = FND_PROFILE.VALUE_SPECIFIC('PER_SECURITY_
PROFILE_ID', URP.USER_ID, URP.RESPONSIBILITY_ID, URP.RESPONSIBILITY_
APPLICATION_ID)
AND PER.SECURITY_PROFILE_ID = PSEC.SECURITY_PROFILE_ID
AND PER.PERSON_ID = ASG.PERSON_ID
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND ASG.EFFECTIVE_END_DATE
AND URP.RESPONSIBILITY_ID = DECODE(FND_GLOBAL.RESP_ID,
          -1, URP.RESPONSIBILITY_ID,
          NULL, URP.RESPONSIBILITY_ID,
          FND_GLOBAL.RESP_ID)
UNION
SELECT DISTINCT 'HR_ORG',
  ORGANIZATION_ID
FROM PER_ALL_ASSIGNMENTS_F ASG,
  FND_USER USR
WHERE ASG.PERSON_ID = USR.EMPLOYEE_ID
AND USR.USER_NAME = UPPER('VALUEOF(NQ_SESSION.USER)')
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND ASG.EFFECTIVE_END_DATE
AND ASG.PRIMARY_FLAG = 'Y'
) SEC_DET
```

The following SQL should be used when MSG is in place:

```
SELECT
  DISTINCT 'HR_ORG',
  TO_CHAR(PER_ORGANIZATION_LIST.ORGANIZATION_ID)
FROM PER_ORGANIZATION_LIST,
  (SELECT FND_PROFILE.VALUE_SPECIFIC('PER_BUSINESS_GROUP_ID', USER_ID,
RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID) PROFILE_ID
FROM
  (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID
FROM FND_USER_RESP_GROUPS
WHERE START_DATE < SYSDATE
AND (CASE WHEN END_DATE IS NULL THEN SYSDATE ELSE TO_DATE(END_DATE) END) >=
SYSDATE
AND USER_ID = (CASE WHEN 'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' =
'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_USER_ID) ELSE (SELECT USER_ID FROM FND_USER
WHERE USER_NAME = UPPER('VALUEOF(NQ_SESSION.USER)')) END)
AND RESPONSIBILITY_ID = (CASE WHEN 'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_
MODE)' = 'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END)
AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN
'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' = 'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE RESPONSIBILITY_APPLICATION_
ID END)
)
WHERE
PER_ORGANIZATION_LIST.SECURITY_PROFILE_ID = PROFILE_ID
```

Note: The 'HR Org-Based Security' application role contains all the data access permission filters. When users create ad-hoc reports, they see the data that is assigned with their permissions. For reports involved with the tables defined above, users are restricted to the data pertaining to their visibility in the organization structure.

2.6.7 Human Resource Personnel Data Analyst Security for Oracle EBS

HR personnel need to see all data for the internal organizations for which they are responsible and the data for their subordinates in their own organization. The 'Human Resource Personnel Data Security' application role supports this requirement. The security mechanism for this application role uses the following metadata elements:

- HR_ORG variable. This variable is defined by the row-wise initialization block HR Organizations. This data set stores all the organizations the user is responsible for, plus the user's own organization, which is the same as the organization selected in USER_HR_ORG. The query for populating this data set is:

  Note: The actual SQL query differs depending on whether Multiple Security Group (MSG) is set up or not.

  The following SQL is used when MSG is not in place:

```
SELECT
 DISTINCT 'HR_ORG',
TO_CHAR(PER_ORGANIZATION_LIST.ORGANIZATION_ID)
FROM PER_ORGANIZATION_LIST,
(SELECT FND_PROFILE.VALUE_SPECIFIC('PER_BUSINESS_GROUP_ID', USER_ID,
RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID)
PROFILE_ID
FROM
(SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID FROM FND_
USER_RESP_GROUPS WHERE START_DATE < SYSDATE AND (CASE WHEN END_DATE IS NULL
THEN SYSDATE ELSE TO_DATE(END_DATE) END) >= SYSDATE
 AND USER_ID = (CASE WHEN 'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' =
'Integrated' THEN VALUEOF(NQ_SESSION.OLTP_EBS_USER_ID) ELSE (SELECT USER_ID
FROM FND_USER WHERE USER_NAME = UPPER('VALUEOF(NQ_SESSION.USER)')) END)
AND RESPONSIBILITY_ID = (CASE WHEN 'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_
MODE)' = 'Integrated'
 THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END)
AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN 'VALUEOF(NQ_SESSION.EBS_SSO_
INTEGRATION_MODE)' = 'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE RESPONSIBILITY_APPLICATION_
ID END)
)
WHERE
PER_ORGANIZATION_LIST.SECURITY_PROFILE_ID = PROFILE_ID
```

  The following SQL is used when MSG is in place:

```
SELECT
 DISTINCT 'HR_ORG'
,TO_CHAR(SEC_DET.ORGANIZATION_ID)
FROM
(
SELECT 'HR_ORG',
ASG.ORGANIZATION_ID
FROM FND_USER_RESP_GROUPS URP,
  FND_USER USR,
  PER_SEC_PROFILE_ASSIGNMENTS SASG,
  PER_SECURITY_PROFILES PSEC,
```

```
  PER_PERSON_LIST PER,
  PER_ALL_ASSIGNMENTS_F ASG
WHERE URP.START_DATE < TRUNC(SYSDATE)
AND (CASE WHEN URP.END_DATE IS NULL THEN TRUNC(SYSDATE) ELSE TO_DATE(URP.END_
DATE) END) >= TRUNC(SYSDATE)
AND USR.USER_NAME = UPPER('VALUEOF(NQ_SESSION.USER)')
AND URP.SECURITY_GROUP_ID = SASG.SECURITY_GROUP_ID
AND URP.USER_ID = USR.USER_ID
AND TRUNC(SYSDATE)
  BETWEEN URP.START_DATE AND NVL(URP.END_DATE, HR_GENERAL.END_OF_TIME)
AND URP.USER_ID = SASG.USER_ID
AND URP.RESPONSIBILITY_ID = SASG.RESPONSIBILITY_ID
AND URP.RESPONSIBILITY_APPLICATION_ID = SASG.RESPONSIBILITY_APPLICATION_ID
AND PSEC.SECURITY_PROFILE_ID = SASG.SECURITY_PROFILE_ID
AND PSEC.SECURITY_PROFILE_ID = PER.SECURITY_PROFILE_ID
AND PER.PERSON_ID = ASG.PERSON_ID
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND ASG.EFFECTIVE_END_DATE
AND TRUNC(SYSDATE) BETWEEN SASG.START_DATE AND NVL(SASG.END_DATE, HR_
GENERAL.END_OF_TIME)
AND URP.RESPONSIBILITY_ID = DECODE(FND_GLOBAL.RESP_ID,
          -1, URP.RESPONSIBILITY_ID,
          NULL, URP.RESPONSIBILITY_ID,
          FND_GLOBAL.RESP_ID)
UNION
SELECT DISTINCT 'HR_ORG', ORGANIZATION_ID
FROM PER_ALL_ASSIGNMENTS_F ASG,
  FND_USER USR
WHERE ASG.PERSON_ID = USR.EMPLOYEE_ID
AND USR.USER_NAME = UPPER('VALUEOF(NQ_SESSION.USER)')
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND ASG.EFFECTIVE_END_DATE
AND ASG.PRIMARY_FLAG= 'Y'
) SEC_DET
```

- USER_HR_ORG variable. This variable is defined using the initialization block User HR Organizations. This variable stores the user's own organization. The query for populating this variable is:

```
SELECT DISTINCT 'USER_HR_ORG', ORGANIZATION_ID
FROM PER_ALL_ASSIGNMENTS_F ASG,
  FND_USER USR
WHERE ASG.PERSON_ID = USR.EMPLOYEE_ID
AND USR.USER_NAME = UPPER('VALUEOF(NQ_SESSION.USER)')
AND TRUNC(SYSDATE) BETWEEN ASG.EFFECTIVE_START_DATE AND ASG.EFFECTIVE_END_DATE
AND ASG.PRIMARY_FLAG= 'Y'
```

- Human Resources Analyst application role. The data filter defined for this application role is the following:

```
Core."Dim - Employee Organization"."Employee Organization
Number" = VALUEOF(NQ_SESSION."HR_ORG") AND (Core."Dim -
Employee Organization"."Employee Organization Number" <>
VALUEOF(NQ_SESSION."USER_HR_ORG") OR Core."Dim - Position Security"."Hierarchy
Based Column" = VALUEOF(NQ_
SESSION."USER"))
```

  This filter joins the fact table used in the report to the Employee Organization dimension to get the organization number for the employee owner of the fact record. If this organization is among the HR orgs, then it will be compared next to the user's own organization. If they are different, then there is no further check, and the record is selected. If they are the same, then an additional filter is applied

based on the employee hierarchy, to make sure the employee owner of this fact record is one of the user's subordinates.

2.6.8 Employee-Based Security for Oracle EBS

Employee-based security restricts data visibility of the records to the owner of that record, and all employees reporting to him or her in the company's employee hierarchy. This security mechanism uses data from the data warehouse database, and shares the metadata components with other supported applications (Siebel CRM and PeopleSoft). By default, this type of security supports only HR Analytics facts. For more information on how this security mechanism works, see Section 2.8.2, "About Primary Position-Based Security for Siebel CRM Industry Applications."

2.7 Integrating Data Security for Oracle's PeopleSoft Enterprise Applications

This section explains how data security is implemented for Oracle's PeopleSoft Enterprise Applications in Oracle BI Applications. Read this section if you want to understand how the default security settings are configured so that you can change the way security is implemented if required. This section contains the following topics:

2.7.1 Oracle BI Applications Authorization for PeopleSoft

The authorization process of Oracle BI Applications fetches a user's role from the source PeopleSoft application, matches the role with all Oracle BI Applications application roles, and determines the user's applicable object security during a user's session. The initialization block 'Authorization' is used to fetch roles and assign the result set to a special session variable called 'GROUP', which the Oracle BI Server then uses for matching. The initialization block SQL is the following:

```
SELECT DISTINCT
'GROUP', ROLENAME
FROM
PSROLEUSER
WHERE
ROLEUSER = ''VALUEOF(NQ_SESSION.USER)''
```

2.7.2 Operating Unit-Based Security for PeopleSoft Financials and PeopleSoft Procurement and Spend

The sequence for operating unit-based security for PeopleSoft Financials and PeopleSoft Procurement and Spend is described below:

1. When a user logs in to Oracle BI Applications, the following session variable is set automatically.

   ```
   USER (System variable)
   ```

2. The Oracle BI Server then gets the operating units (or the general ledger business units in PeopleSoft Financials) corresponding to the USER from the following tables:

   – PS_SEC_BU_OPR

   – PS_BUS_UNIT_TBL_GL

   – PS_INSTALLATION_FS

   – PS_SEC_BU_CLS

   The following session variable is set automatically:

   ```
   OU_ORG (Row-wise variable)
   ```

   The initialization block 'Operating Unit Organizations', which sets the value for this variable, is shown below.

   Initialization block -- 'Operating Unit Organizations'

   The initialization block 'Operating Unit Organizations' sets value for variable OU_ORG using the following SQL:

   ```
   SELECT DISTINCT 'OU_ORG', S1.BUSINESS_UNIT
   FROM PS_SEC_BU_OPR S1, PS_BUS_UNIT_TBL_GL A, PS_INSTALLATION_
   FS I
   WHERE S1.OPRID = ''VALUEOF(NQ_SESSION.USER)''
   AND S1.BUSINESS_UNIT = A.BUSINESS_UNIT
   AND I.SECURITY_TYPE = 'O'
   AND I.BU_SECURITY = 'Y'
   UNION
   SELECT DISTINCT 'OU_ORG', S2.BUSINESS_UNIT
   FROM PS_SEC_BU_CLS S2,
   PS_BUS_UNIT_TBL_GL A,
   PS_INSTALLATION_FS I2,
   PSOPRDEFN P
   WHERE P.OPRID = ''VALUEOF(NQ_SESSION.USER)''
   AND S2.BUSINESS_UNIT = A.BUSINESS_UNIT
   AND P.OPRCLASS = S2.OPRCLASS
   AND I2.SECURITY_TYPE = 'C'
   AND I2.BU_SECURITY = 'Y'
   ```

Note: The 'Operating Unit Org-Based Security' application role contains all the data access permission filters.

2.7.3 Company Org-Based Security for PeopleSoft Financials and PeopleSoft HR

The sequence for company org-based security for PeopleSoft Financials and PeopleSoft HR is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

   ```
   USER (System variable)
   ```

2. The Oracle BI Server then gets the companies or business units corresponding to the USER from the following tables:
   - PS_SEC_BU_OPR
   - PS_BUS_UNIT_TBL_GL
   - PS_SCRTY_TBL_DEPT
   - PS_BU_DEPT_VW
   - PS_BUS_UNIT_TBL_GL
   - PSOPRDEFN for PeopleSoft HR
   - PS_INSTALLATION_FS for PeopleSoft Financials
   - PSOPRDEFN for PeopleSoft Financials
   - PS_SEC_BU_CLS for PeopleSoft Financials

   The following session variable is set automatically:

   ```
   COMPANY (Row-wise variable)
   ```

   The initialization block 'Companies', which sets the value for this variable, is shown below.

   Initialization block -- 'Companies'

   The initialization block 'Companies' sets value for variable COMPANY using the following SQL:

   For PeopleSoft Financials:

   ```
   SELECT DISTINCT 'COMPANY', S1.BUSINESS_UNIT
   FROM PS_SEC_BU_OPR S1,PS_BUS_UNIT_TBL_GL A, PS_INSTALLATION_
   FS I
   WHERE S1.OPRID = 'VALUEOF(NQ_SESSION.USER)'
   AND S1.BUSINESS_UNIT = A.BUSINESS_UNIT
   AND I.SECURITY_TYPE = 'O'
   UNION
   SELECT DISTINCT 'COMPANY', S2.BUSINESS_UNIT
   FROM PS_SEC_BU_CLS S2,
   PS_BUS_UNIT_TBL_GL A,
   PS_INSTALLATION_FS I2,
   PSOPRDEFN P
   ```

```
WHERE P.OPRID = 'VALUEOF(NQ_SESSION.USER)'

AND S2.BUSINESS_UNIT = A.BUSINESS_UNIT

AND P.OPRCLASS = S2.OPRCLASS

AND I2.SECURITY_TYPE = 'C'

AND I2.BU_SECURITY = 'Y'
```

For PeopleSoft HR:

```
SELECT DISTINCT 'COMPANY',C.BUSINESS_UNIT

FROM PSOPRDEFN A, PS_SCRTY_TBL_DEPT B, PS_BU_DEPT_VW C, PS_
BUS_UNIT_TBL_GL D

WHERE

A.ROWSECCLASS = B.ROWSECCLASS AND

B.ACCESS_CD = 'Y' AND

B.DEPTID = C.DEPTID AND

C.BUSINESS_UNIT = D.BUSINESS_UNIT AND

A.OPRID = 'VALUEOF(NQ_SESSION.USER)'
```

Note: The 'Company Org-Based Security' application role contains all the data access permission filters.

2.7.4 Ledger-Based Security for PeopleSoft Financials and PeopleSoft Procurement and Spend

Ledger data in PeopleSoft is reference data that is secured by and shared by business units. The Ledger table includes the SetID field and uses the TableSet feature in PeopleTool. In addition, ledger data access is controlled by row-level security, which enables you to implement security to restrict individual users or permission lists from specific rows of data that are controlled by the ledger. Ledger-based security filters data based on the ledgers associated with the user that is logged in.

When you set up ledger-based security for a PeopleSoft application, you should also set up the company org-based security for PeopleSoft. Ledger-based security does not automatically restrict the data by the GL business unit.

The sequence for ledger-based security for PeopleSoft Financials and PeopleSoft Procurement and Spend is described below:

1. When a user logs in to Oracle BI Applications, the following session variable is set automatically:

   ```
   USER (System variable)
   ```

2. The Oracle BI Server gets the ledgers corresponding to the USER from the following tables:

   - PS_LED_DEFN_TBL
   - PS_INSTALLATION_FS
   - PS_SEC_LEDGER_CLS
   - PS_LED_GRP_TBL
   - PSOPRDEFN

- PSROLEUSER
- PSROLECLASS

The following session variable is set automatically:

```
LEDGER (Row-wise variable)
```

The initialization block 'Ledgers', which sets the value for this variable, is set as follows.

```
SELECT DISTINCT'LEDGER', LG.SETID || SO.LEDGER

FROM PS_SEC_LEDGER_OPR SO, PS_LED_DEFN_TBL LG, PS_
INSTALLATION_FS IFS

WHERE SO.LEDGER = LG.LEDGER AND IFS.SECURITY_TYPE = 'O'

AND IFS.LEDGER_SECURITY = 'Y' AND SO.OPRID = 'VALUEOF(NQ_
SESSION.USER)'

UNION

SELECT distinct 'LEDGER', LG.SETID || SC.LEDGER

FROM PS_SEC_LEDGER_CLS SC, PS_LED_GRP_TBL LG, PSOPRDEFN OP,
PSROLEUSER ORL, PSROLECLASS RCL, PS_INSTALLATION_FS IFS

WHERE SC.LEDGER_GROUP = LG.LEDGER_GROUP AND SC.OPRCLASS =
RCL.CLASSID AND OP.OPRID = ORL.ROLEUSER

AND ORL.ROLENAME = RCL.ROLENAME and IFS.SECURITY_TYPE = 'C'
AND IFS.LEDGER_SECURITY = 'Y' AND OP.OPRID = 'VALUEOF(NQ_
SESSION.USER)'
```

Note: The 'Ledger-Based Security' application role contains all the data access permission filters.

2.7.5 HR Org-Based Security for PeopleSoft HR

HR org-based security for PeopleSoft HR supports the PeopleSoft department security by tree.

The sequence for HR org-based security with PeopleSoft HR is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

   ```
   USER (System variable)
   ```

2. The Oracle BI Server gets the HR business units corresponding to the USER from the following tables:

   – PSOPRDEFN

   – PS_SCRTY_TBL_DEPT

   The following session variable is set automatically:

   ```
   HR_ORG (Row-wise variable)
   ```

   The initialization block 'HR Organizations', which sets the value for this variable, is shown below.

   Initialization block -- 'HR Organizations'

   The initialization block 'HR Organizations' sets value for variable HR_ORG using the following SQL:

```
SELECT 'DEPT_ID', DEPT.DEPTID
FROM
PS_DEPT_TBL DEPT,PSOPRDEFN OPR
WHERE DEPT.EFFDT = (
SELECT MAX(DEPT1.EFFDT)
FROM PS_DEPT_TBL DEPT1
WHERE DEPT1.SETID = DEPT.SETI
AND DEPT1.DEPTID = DEPT.DEPTID)
AND (EXISTS (
SELECT 'X'
FROM PS_SJT_DEPT SEC,
PS_SJT_CLASS_ALL CLS,
PS_SJT_OPR_CLS SOC
WHERE SEC.SETID = DEPT.SETID
AND SEC.DEPTID = DEPT.DEPTID
AND CLS.SCRTY_SET_CD = 'PPLJOB'
AND CLS.SCRTY_TYPE_CD = '001'
AND CLS.TREE = 'Y'
AND CLS.SCRTY_KEY1 = SEC.SCRTY_KEY1
AND CLS.SCRTY_KEY2 = SEC.SCRTY_KEY2
AND CLS.SCRTY_KEY3 = SEC.SCRTY_KEY3
AND SOC.OPRID = OPR.OPRID
AND SOC.CLASSID = CLS.CLASSID
AND SOC.CLASSID = OPR.ROWSECCLASS
AND SOC.SEC_RSC_FLG <> '2' )
OR EXISTS (
SELECT 'X'
FROM PS_SJT_DEPT SEC,
PS_SJT_CLASS_ALL CLS,
PS_SJT_OPR_CLS SOC
WHERE SEC.SETID = DEPT.SETID
AND SEC.DEPTID = DEPT.DEPTID
AND CLS.SCRTY_SET_CD = 'DEPT'
AND CLS.SCRTY_TYPE_CD = SEC.SCRTY_TYPE_CD
AND CLS.SCRTY_KEY1 = SEC.SCRTY_KEY1
AND CLS.SCRTY_KEY2 = SEC.SCRTY_KEY2
AND CLS.SCRTY_KEY3 = SEC.SCRTY_KEY3
AND CLS.TREE = 'Y'
AND SOC.OPRID = OPR.OPRID
AND SOC.CLASSID = CLS.CLASSID
AND SOC.CLASSID = OPR.ROWSECCLASS
AND SOC.SEC_RSC_FLG <> '2' )
OR EXISTS (
SELECT 'X'
FROM PS_SJT_DEPT SEC,
PS_SJT_CLASS_ALL CLS,
PS_SJT_OPR_CLS SOC
WHERE SEC.SETID = DEPT.SETID
AND SEC.DEPTID = DEPT.DEPTID
AND CLS.SCRTY_SET_CD = 'DEPT'
AND CLS.SCRTY_TYPE_CD = SEC.SCRTY_TYPE_CD
AND CLS.SCRTY_KEY1 = SEC.SCRTY_KEY1
AND CLS.SCRTY_KEY2 = SEC.SCRTY_KEY2
AND CLS.SCRTY_KEY3 = SEC.SCRTY_KEY3
AND CLS.TREE = 'N'
AND SOC.OPRID = OPR.OPRID
AND SOC.CLASSID = CLS.CLASSID))
AND OPR.OPRID = 'VALUEOF(NQ_SESSION.USER)'
```

Note: The 'HR Org-Based Security' application role contains all the data access permission filters. When users create ad-hoc reports, they see the data that is assigned with their permissions. For reports involved with the tables defined above, users are restricted to the data pertaining to their visibility in the organization structure.

2.7.6 Payables Org-Based Security for PeopleSoft Financials

The sequence for payables org-based security for PeopleSoft Financials is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

   ```
   USER (System variable)
   ```

2. The Oracle BI Server gets the Payables business units corresponding to the USER from the following tables:

   - PSOPRDEFN
   - PS_SEC_BU_OPR
   - PS_SEC_BU_CLS
   - PS_INSTALLATION_FS
   - PS_BUS_UNIT_TBL_AP

   The following session variable is set automatically:

   ```
   PAYABLES_ORG (Row-wise variable)
   ```

   The initialization block 'Payables Organizations', which sets the value for this variable, is shown below.

   Initialization block -- 'Payables Organizations'

   The initialization block 'Payables Organizations' sets value for variable PAYABLES_ORG using the following SQL:

   ```
   SELECT DISTINCT 'PAYABLES_ORG', s1.BUSINESS_UNIT
   FROM PS_SEC_BU_OPR s1, PS_BUS_UNIT_TBL_AP a, PS_INSTALLATION_
   FS i
   WHERE s1.OPRID = 'VALUEOF(NQ_SESSION.USER)'
   AND s1.BUSINESS_UNIT = a.BUSINESS_UNIT
   AND i.SECURITY_TYPE = 'O'
   AND i.BU_SECURITY = 'Y'
   UNION
   SELECT DISTINCT 'PAYABLES_ORG', s2.BUSINESS_UNIT
   FROM PS_SEC_BU_CLS s2, PS_BUS_UNIT_TBL_AP a, PS_INSTALLATION_
   FS i2, PSOPRDEFN p
   WHERE p.OPRID = 'VALUEOF(NQ_SESSION.USER)'
   AND s2.BUSINESS_UNIT = a.BUSINESS_UNIT
   AND p.OPRCLASS = s2.OPRCLASS
   AND i2.SECURITY_TYPE = 'C'
   AND i2.BU_SECURITY = 'Y'
   ```

Note: The 'Payables Org-Based Security' application role contains all the data access permission filters. When users create ad-hoc reports, they see the data that is assigned with their permissions. For reports involved with the tables defined above, users are restricted to the data pertaining to their visibility in the organization structure.

2.7.7 Receivables Org-Based Security for PeopleSoft Financials

The sequence for receivables org-based security for PeopleSoft Financials is described below:

1. When a user logs in to Oracle BI Applications, the session variable below is set automatically.

   ```
   USER (System variable)
   ```

2. The Oracle BI Server gets the Receivables business units corresponding to the USER from the following tables:

   – PS_SEC_BU_OPR

   – PS_SEC_BU_CLS

   – PS_INSTALLATION_FS

   – PS_BUS_UNIT_TBL_AR

   The following session variable is set automatically:

   ```
   RECEIVABLES_ORG (Row-wise variable)
   ```

   The initialization block 'Receivables Organizations', which sets the value for this variable, is shown below.

   Initialization block -- 'Receivables Organizations'

   The initialization block 'Receivables Organizations' sets value for variable RECEIVABLES_ORG using the following SQL:

   ```
   SELECT DISTINCT 'RECEIVABLES_ORG', s1.BUSINESS_UNIT

   FROM PS_SEC_BU_OPR s1, PS_BUS_UNIT_TBL_AR a, PS_INSTALLATION_
   FS i

   WHERE s1.OPRID = 'VALUEOF(NQ_SESSION.USER)'

   AND s1.BUSINESS_UNIT = a.BUSINESS_UNIT AND i.SECURITY_TYPE =
   'O'

   AND i.BU_SECURITY = 'Y'

   UNION

   SELECT DISTINCT 'RECEIVABLES_ORG', s2.BUSINESS_UNIT

   FROM PS_SEC_BU_CLS s2, PS_BUS_UNIT_TBL_AR a, PS_INSTALLATION_
   FS i2, PSOPRDEFN p

   WHERE p.OPRID = 'VALUEOF(NQ_SESSION.USER)'

   AND s2.BUSINESS_UNIT = a.BUSINESS_UNIT AND p.OPRCLASS =
   s2.OPRCLASS AND i2.SECURITY_TYPE = 'C'

   AND i2.BU_SECURITY = 'Y'
   ```

Note: The 'Receivables Org-Based Security' application role contains all the data access permission filters. When users create ad-hoc reports, they see the data that is assigned

with their permissions. For reports involved with the tables defined above, users are restricted to the data pertaining to their visibility in the organization structure.

2.7.8 SetID-Based Security for PeopleSoft HR, PeopleSoft Financials, and PeopleSoft Procurement and Spend

The sequence for SetID-based security for PeopleSoft Financials, PeopleSoft HR, and PeopleSoft Procurement and Spend is described below:

1. When a user logs in to Oracle BI Applications, the following session variable is set automatically:

   ```
   USER (System variable)
   ```

2. The Oracle BI Server gets the SetIDs corresponding to the USER from the following tables:

   - PS_SEC_SETID_OPR
   - PS_SEC_SETID_CLS
   - PS_INSTALLATION_FS
   - PSOPRDEFN

   The following session variable is set automatically:

   ```
   SET_ID (Row-wise variable)
   ```

   The initialization block 'Set ID' sets value for variable SET_ID using the following SQL:

   For PeopleSoft Financials:

   ```
   SELECT DISTINCT 'SET_ID', s1.SETID
   FROM PS_SEC_SETID_OPR s1, PS_INSTALLATION_FS i
   WHERE s1.OPRID = 'VALUEOF(NQ_SESSION.USER)'
   AND i.SECURITY_TYPE = 'O'
   AND i.SETID_SECURITY = 'Y'
   UNION
   SELECT DISTINCT 'SET_ID', s2.SETID
   FROM PS_SEC_SETID_CLS s2, PS_INSTALLATION_FS i2, PSOPRDEFN p
   WHERE p.OPRID = 'VALUEOF(NQ_SESSION.USER)'
   AND p.OPRCLASS = s2.OPRCLASS
   AND i2.SECURITY_TYPE = 'C'
   AND i2.SETID_SECURITY = 'Y'
   ```

Note: The 'Set ID-Based Security' application role contains all the data access permission filters.

2.7.9 Human Resource Personnel Data Analyst Security for PeopleSoft HR

HR personnel need to see all data for the internal organizations for which they are responsible for and the data for their subordinates in their own organization. The 'Human Resource Personnel Data Security' application role supports this requirement.

The security mechanism for this application role uses the following metadata elements:

- HR_ORG variable. This variable is defined by the row-wise initialization block HR Organizations. This data set stores all the organizations the user is responsible for, plus the user's own organization, which is the same organization as the one selected in USER_HR_ORG. The query for populating this data set is the following:

```
SELECT DISTINCT

'HR_ORG',

C.BUSINESS_UNIT

FROM

PSOPRDEFN A, PS_SCRTY_TBL_DEPT B, PS_BU_DEPT_VW C, PS_BUS_
UNIT_TBL_HR D

WHERE

A.ROWSECCLASS = B.ROWSECCLASS AND

B.ACCESS_CD = 'Y' AND

B.DEPTID = C.DEPTID AND

C.BUSINESS_UNIT = D.BUSINESS_UNIT AND

A.OPRID = 'VALUEOF(NQ_SESSION.USER)'

UNION

SELECT DISTINCT 'HR_ORG', FINAL_JOB.BUSINESS_UNIT

FROM (

SELECT X.EMPLID, MAX(X.BUSINESS_UNIT) BUSINESS_UNIT FROM

(

SELECT A.EMPLID, A.EMPL_RCD, A.EFFDT,EFFSEQ, A.JOB_
INDICATOR,A.EMPL_STATUS, A.BUSINESS_UNIT

FROM PS_JOB A,

(SELECT EMPLID, MAX(EFFDT) MAX_EFFDT

FROM PS_JOB

WHERE

JOB_INDICATOR = 'P' AND EMPL_STATUS IN ('A', 'L', 'P', 'W')

GROUP BY EMPLID) B

WHERE

A.EMPLID = B.EMPLID

AND A.EFFDT = B.MAX_EFFDT

AND A.JOB_INDICATOR = 'P' AND A.EMPL_STATUS IN ('A', 'L',
'P', 'W')

AND A.EFFSEQ = (SELECT MAX (C.EFFSEQ)

FROM PS_JOB C

WHERE
```

```
C.EMPLID = A.EMPLID AND
C.EMPL_RCD = A.EMPL_RCD AND
C.EFFDT = A.EFFDT AND
C.JOB_INDICATOR = 'P' AND C.EMPL_STATUS IN ('A', 'L', 'P',
'W'))
) X
GROUP BY X.EMPLID
) FINAL_JOB, PSOPRDEFN
WHERE
FINAL_JOB.EMPLID = PSOPRDEFN.EMPLID AND
PSOPRDEFN.OPRID = 'VALUEOF(NQ_SESSION.USER)'
```

- USER_HR_ORG variable. This variable is defined using the initialization block User HR Organizations. This variable stores the user's own organization. The query for populating this variable is the following:

```
SELECT DISTINCT FINAL_JOB.BUSINESS_UNIT
FROM (
SELECT X.EMPLID, MAX(X.BUSINESS_UNIT) BUSINESS_UNIT FROM
(
SELECT A.EMPLID, A.EMPL_RCD, A.EFFDT,EFFSEQ, A.JOB_INDICATOR,
A.EMPL_STATUS, A.BUSINESS_UNIT
FROM PS_JOB A,
(SELECT EMPLID, MAX(EFFDT) MAX_EFFDT
FROM PS_JOB
WHERE
JOB_INDICATOR = 'P' AND EMPL_STATUS IN ('A', 'L', 'P', 'W')
GROUP BY EMPLID) B
WHERE
A.EMPLID = B.EMPLID
AND A.EFFDT = B.MAX_EFFDT
AND A.JOB_INDICATOR = 'P' AND A.EMPL_STATUS IN ('A', 'L',
'P', 'W')
AND A.EFFSEQ = (SELECT MAX (C.EFFSEQ)
FROM PS_JOB C
 WHERE
C.EMPLID = A.EMPLID AND
C.EMPL_RCD = A.EMPL_RCD AND
C.EFFDT = A.EFFDT AND
C.JOB_INDICATOR = 'P' AND C.EMPL_STATUS IN ('A', 'L', 'P',
'W'))
```

```
) X

GROUP BY X.EMPLID

) FINAL_JOB, PSOPRDEFN

WHERE

FINAL_JOB.EMPLID = PSOPRDEFN.EMPLID AND

PSOPRDEFN.OPRID = 'VALUEOF(NQ_SESSION.USER)'
```

- Human Resources Analyst application role. The data filter defined for this application role is the following:

```
Core."Dim - Employee Organization"."Employee Organization
Number" = VALUEOF(NQ_SESSION."HR_ORG") AND (Core."Dim -
Employee Organization"."Employee Organization Number" <>
VALUEOF(NQ_SESSION."USER_HR_ORG") OR Core."Dim - Position
Security"."Hierarchy Based Column" = VALUEOF(NQ_
SESSION."USER"))
```

This filter joins the fact table used in the report to the Employee Organization dimension to get the organization number for the employee owner of the fact record. If this organization is among the HR orgs, then it will be compared next to the user's own organization. If they are different, then there is no further check, and the record is selected. If they are the same, then an additional filter is applied based on the employee hierarchy, to make sure the employee owner of this fact record is one of the user's subordinates.

2.7.10 Employee-Based Security for PeopleSoft

Employee-based security restricts data visibility of the records to the owner of that record, and all employees who report to him or her in the company's employee hierarchy. This security mechanism uses data from the Oracle Business Analytics Warehouse database, and shares the metadata components with other supported applications (for example, Oracle EBS, Siebel CRM, or PeopleSoft). By default, this type of security supports only HR Analytics facts. For more information about how this security mechanism works, see Section 2.8.2, "About Primary Position-Based Security for Siebel CRM Industry Applications."

2.8 Integrating Data Security for Oracle's Siebel CRM Applications

This section explains how data security in Oracle BI Applications is deployed with Siebel CRM. Read this section if you want to understand how the default security settings are configured so that you can change the way security is implemented if required.

> **Note:** Primary Employee/Position Hierarchy-Based security is not available for Siebel Service Analytics. The security available for Siebel Service Analytics is visibility granted to the primary owner organization.

This section contains the following topics:

2.8.1 About Primary Position-Based Security

This section covers primary position-based security. It contains the following topics:

2.8.1.1 Introduction

Primary position-based security restricts data visibility for a fact or dimension record to the primary owner of this record and those above him in the hierarchy. The primary owner of a record could be a position or an employee. Primary position-based security uses a flattened hierarchy table called W_POSTION_DH, which is based on W_POSITION_D and is treated as a slowly changing dimension.

For Siebel CRM-based data, W_POSITION_D is populated from the Position table in Siebel CRM. A new record is created for the same position every time a new employee is associated with this position as the primary employee.

Consequently, every record in the source tables can be represented by more than one record in W_POSITION_DH, but only one record can have the value of CURRENT_FLG as 'Y' at any time. The W_POSITION_DH table also contains one set of columns prefixed with CURRENT, and another set of columns not prefixed with CURRENT. The columns that are prefixed with CURRENT reflect the current hierarchy structure for the position or employee record at any time. The columns that are not prefixed with CURRENT reflect the hierarchy structure for the same position or employee record during the period between EFFECTIVE_START_DT and EFFECTIVE_END_DT. This latter set of columns is used to enable fact records to be visible to the owner of a record and his upper level managers at the time the record was created, even after he changes position or managers in the company hierarchy.

Facts join to this dimension by the record owner; for example, W_REVN_F is joined using PR_POSITION_DH_WID, where PR_POSITION_DH_WID is the primary position on the revenue line in the source application.

2.8.1.2 Primary Employee/Position Hierarchy-Based Application Role

This application role uses the following metadata elements in the repository:

- HIER_LEVEL session variable. This variable is populated by the initialization block 'User Hierarchy Level' using the SQL below. For a description of the User Hierarchy Level initialization block, see Section 2.2.3, "Initialization Blocks Used for Data-Level Security in Oracle BI Applications."

  ```
  Select round(FIXED_HIER_LEVEL) FROM VALUEOF(OLAPTBO).W_
  POSITION_DH WHERE BASE_LOGIN= 'VALUEOF(NQ_SESSION.USER)' AND
  CURRENT_FLG='Y'
  ```

  The HIER_LEVEL value can be a number between 0 and 17. It designates the current Fixed Hierarchy level of the user in the company hierarchy. The Company

hierarchy is based on the Employee hierarchy tree for Oracle EBS and PeopleSoft applications and on the Position hierarchy tree for Siebel Applications. For example, the CEO of the company is the only employee whose HIER_LEVEL takes the value 17, if the employee hierarchy is a full tree.

- Dim - Position Security logical dimension. This logical dimension is joined to the supported fact tables. It is defined on the physical table W_POSITION_DH.
  - Hierarchy-Based Column logical column. This column is a logical column in the Dim - Position Security logical dimension. It is defined as follows:

```
"INDEXCOL(VALUEOF(NQ_SESSION."HIER_LEVEL"), "Core"."Dim -
Position Security"."Current Base Level Login", "Core"."Dim -
Position Security"."Current Level 1 Login", "Core"."Dim -
Position Security"."Current Level 2 Login", "Core"."Dim -
Position Security"."Current Level 3 Login", "Core"."Dim -
Position Security"."Current Level 4 Login", "Core"."Dim -
Position Security"."Current Level 5 Login", "Core"."Dim -
Position Security"."Current Level 6 Login", "Core"."Dim -
Position Security"."Current Level 7 Login", "Core"."Dim -
Position Security"."Current Level 8 Login", "Core"."Dim -
Position Security"."Current Level 9 Login", "Core"."Dim -
Position Security"."Current Level 10 Login", "Core"."Dim -
Position Security"."Current Level 11 Login", "Core"."Dim -
Position Security"."Current Level 12 Login", "Core"."Dim -
Position Security"."Current Level 13 Login", "Core"."Dim -
Position Security"."Current Level 14 Login", "Core"."Dim -
Position Security"."Current Level 15 Login", "Core"."Dim -
Position Security"."Current Level 16 Login", "Core"."Dim -
Position Security"."Current Top Level Login")".
```

- The IndexCol function in this definition makes the Hierarchy-Based Column default to one of the logical columns in the list based on the value of HIER_LEVEL. So, if the value of HIER_LEVEL is 0, the new column will default to the first column in the list, and so on.
- A filter in the application role 'Primary Employee/Position Hierarchy-Based Security' defined as follows: ("Core"."Dim - Position Security"."Hierarchy Based Column" = VALUEOF(NQ_SESSION."USER")).

  A user needs to be a member of the application role 'Primary Employee/Position Hierarchy-Based Security', through one of his responsibilities (for Siebel and Oracle EBS applications) and Roles (for PeopleSoft applications), for the data security filters to apply. Users are assigned to this application role based on their responsibilities, using the Authorization initialization block, as described in Section 2.2.3, "Initialization Blocks Used for Data-Level Security in Oracle BI Applications." By default, this initialization block is populated using the following SQL:

```
Select 'GROUP', R.NAME

from VALUEOF(TBO).S_RESP R, VALUEOF(TBO).S_PER_RESP P,
VALUEOF(TBO).S_USER U

where U.LOGIN=Upper('VALUEOF(NQ_SESSION.USER)') and U.ROW_
ID=P.PER_ID and P.RESP_ID=R.ROW_ID

UNION

select 'GROUP', CASE VALUEOF(NQ_SESSION.HIER_LEVEL)
```

```
WHEN 0 THEN 'Hierarchy Level (Base)'
 when 1 then 'Hierarchy Level 1'
when 2 then 'Hierarchy Level 2'
when 3 then 'Hierarchy Level 3'
when 4 then 'Hierarchy Level 4'
when 5 then 'Hierarchy Level 5'
when 8 then 'Hierarchy Level 8'
when 6 then 'Hierarchy Level 6'
when 7 then 'Hierarchy Level 7'
when 8 then 'Hierarchy Level 8'
when 9 then 'Hierarchy Level 9'
when 10 then 'Hierarchy Level 10'
when 11 then 'Hierarchy Level 11'
when 12 then 'Hierarchy Level 12'
when 13 then 'Hierarchy Level 13'
when 14 then 'Hierarchy Level 14'
when 15 then 'Hierarchy Level 15'
when 16 then 'Hierarchy Level 16'
When 17 then 'Hierarchy Level (Top)'
ELSE'NOGROUP'END from VALUEOF(TBO).S_DUAL
```

The first part of this SQL selects the user's responsibilities from the Siebel CRM application. The user will be assigned automatically to the application roles with the same name.

The second part of this SQL assigns the user to one of the Oracle BI-specific application roles, such as Hierarchy Level (Base), Hierarchy Level 1 through 16, and Hierarchy Level (Top), based on the variable HIER_LEVEL. These application roles are not used for data security purposes; they are used for presentation column purposes, in conjunction with the Web Choose function defined in some reports. The purpose of this function is to allow a multi-user report to show different position columns to the user, based on his hierarchy level. This is very similar to the IndexCol function described in Section 2.8.1.2, "Primary Employee/Position Hierarchy-Based Application Role."

2.8.1.3 Configuring Oracle BI Repository Table Joins for Primary Employee/Position Hierarchy-Based Security

The procedures below provide instructions for adding primary position-based security to a new dimension or fact table. The following procedures use the W_AGREE_D (Agreement) dimension as an example.

To add primary position-based security to a dimension table:

1. In the Physical layer of the Administration Tool, create an alias on W_POSITION_DH specifically to join to the underlying physical table.
2. Configure the join in the physical layer.

3. In the Business Model and Mapping layer of the Administration Tool, add the W_POSITION_DH alias to the dimension's logical table source.
4. Add new logical columns CURRENT_BASE_LOGIN, CURRENT_LVL1ANC_LOGI, and so on, to the logical table, and map them to the corresponding physical columns.
5. Add the Hierarchy column 'Hierarchy Based Column.'
6. In the Administration Tool, open the Security Manager by selecting Tools and then Identity from the menu bar.
    a. Right-click the application role 'Primary Employee/Position Hierarchy-Based Security,' and choose **Properties**.
    b. In the Properties dialog, click **Permissions**, and select the Data Filters tab.
    c. To add a new filter, click **Add**.
    d. In the new dialog, select the Business Model tab, and find and double-click the logical table Dim - Agreement.

       A new record will be added to the list of filters automatically.
    e. Select the Data Filter field and click the Expression Builder button. Then, add the filter condition "Core"."Dim - Customer"."Hierarchy Based Login" = VALUEOF(NQ_SESSION."USER") in Expression Builder and click **OK**.
    f. Click **OK** in the User/Application Role Permissions dialog, and then click **OK** in the Application Role dialog.

To add primary position-based security support to a fact table:

1. In the Physical layer of the Administration Tool, join the underlying physical table to Dim_W_POSITION_DH_Position_Hierarchy.

   This assumes you already created the appropriate foreign key in the fact table and populated it correctly.
2. Join the logical table to the Dim - Position Security.
3. In the Administration Tool, open the Security Manager by selecting Tools and then Identity from the menu bar.
    a. Right-click the application role 'Primary Employee/Position Hierarchy-based Security', and choose **Properties**.
    b. In the Properties dialog, click **Permissions**, and select the Data Filters tab.
    c. To add a new filter, click **Add**.
    d. In the new dialog box, select the Business Model tab, and find and double-click the logical table: Dim - Agreement.

       A new record will be added to the list of filters automatically.
    e. Select the Data Filter field and click the Expression Builder button. Then, add the condition "Core"."Dim - Position Security"."Hierarchy Based Column" = VALUEOF(NQ_SESSION."USER") in Expression Builder and click **OK**.
    f. Click **OK** in the User/Application Role Permissions dialog, and then click **OK** in the Application Role dialog.

2.8.2 About Primary Position-Based Security for Siebel CRM Industry Applications

This section covers primary position-based security for CRM Industry Applications. It contains the following topics:

- Section 2.8.2.1, "Consumer Sector Analytics Security Settings"
- Section 2.8.2.2, "Communications, Media, and Energy (CME) Analytics Security Settings"
- Section 2.8.2.3, "Financial Services Analytics Security Settings"
- Section 2.8.2.4, "Pharma Sales Analytics and Pharma Marketing Analytics Security Settings"

2.8.2.1 Consumer Sector Analytics Security Settings

Table 2–4 describes the consumer sector responsibilities associated with each CS Dashboard.

Table 2–4 Consumer Sector Responsibilities Associated with Each CS Dashboard

Responsibility	Dashboard	Pages
VP Sales	1. VP Sales 2. Sales Performance 3. Promotion	1. Business Overview, Product Overview 2. Sales Volume Planning, Hierarchy, Trends, Growth 3. Plan Year to Date, Corporate
Key Account Manager	1. Key Account Manager 2. Promotion 3. Funds 4. Retail Audit 5. Sales Performance	1. Business, Category 2. Plan year to date, Key account 3. Account 4. Last audit, Trends 5. Sales Volume Planning, Hierarchy, Trends, Growth

2.8.2.2 Communications, Media, and Energy (CME) Analytics Security Settings

Oracle's CME family of products (Oracle Communications, Media and Energy Sales Analytics, Oracle Communications, Media and Energy Service Analytics, Oracle Communications, Media and Energy Marketing Analytics) use the Siebel operational applications security model; that is, it uses Siebel operational applications responsibilities (and corresponding application roles) for controlling access to Siebel operational applications objects (both metadata and Presentation Services objects). This security model is described in the topic Section 2.1, "About Security in Oracle BI Applications."

In addition to responsibilities provided by the operational applications, Oracle Communications, Media, and Energy (CME) provides additional responsibilities, and responsibility-specific security, as indicated in Table 2–5.

Table 2–5 CME Responsibilities Associated with Each CME Dashboard

CME Responsibility	CME Dashboard	Dashboard Pages
CM Marketing Analytics User CM Marketing Analytics Administrator	Loyalty Management	▪ Customer Lifetime Value ▪ Churn Propensity ▪ Selling Propensity ▪ Financial Risk ▪ Actual Churn

Table 2–5 (Cont.) CME Responsibilities Associated with Each CME Dashboard

CME Responsibility	CME Dashboard	Dashboard Pages
CM Sales Analytics User CM Sales Analytics Administrator	Revenue Management	▪ Sales Portal ▪ Service Activations ▪ Service Modifications ▪ Service Disconnections
-	Account Management	▪ Sales Portal ▪ Service Activations ▪ Service Modifications ▪ Service Disconnections
CM Service Analytics User CM Service Analytics Administrator	Account Management	▪ Trouble Tickets ▪ Customer Satisfaction

2.8.2.3 Financial Services Analytics Security Settings

The following applications use the Siebel operational applications security model:

- The Financial Analytics family of products (Finance Sales Analytics, Finance Service Analytics, Finance Marketing Analytics, Finance Institutional Analytics, Finance Retail Analytics).
- The Insurance Analytics family of products (Insurance Partner Manager Analytics, Insurance Sales Analytics, Insurance Service Analytics, Insurance Marketing Analytics, Insurance Partner Manager Analytics).

In addition to responsibilities provided by the Siebel operational applications, these applications provide additional responsibilities, and responsibility-specific security, as indicated in Table 2–6.

For the Financial Services products, the Siebel operational applications security model has been extended in the following ways:

- **Financial Analytics user**

  A finance-specific responsibility (and corresponding application role) that must be used in conjunction with Siebel operational applications responsibilities and groups to control access to Finance-specific objects in Financial Analytics.

- A user in the Insurance Analytics family of products (Insurance Partner Manager Analytics, Insurance Sales Analytics, Insurance Service Analytics, Insurance Marketing Analytics, Insurance Partner Manager Analytics)

  An insurance-specific responsibility (and corresponding application role) that must be used to control access to the Insurance and Healthcare-specific objects in Insurance and the Healthcare Analytics family of products (Healthcare Sales Analytics, Healthcare Service Analytics, Healthcare Marketing Analytics, Healthcare Partner Manager Analytics).

For example, when you give a salesperson all horizontal Sales responsibilities and also include the finance responsibility Financial Analytics User, this user is able to see, in addition to all horizontal sales objects (dashboards, subject areas, folders in the Presentation layer, and so on), all finance-specific Sales objects. Similarly, in order to see Insurance and Healthcare-specific objects, you need to add one of the Insurance Analytics family of products (Insurance Partner Manager Analytics, Insurance Sales Analytics, Insurance Service Analytics, Insurance Marketing Analytics, Insurance Partner Manager Analytics) user responsibilities to this user.

2.8.2.3.1 Parent and Child Application Role Behavior Oracle BI Applications supports hierarchies in application roles, and certain application roles within the policy store are parent application roles that define the behavior of all the child application roles. For Financial Services Analytics, the parent application roles are the following:

- **Finance**

  Parent application role for all application roles for Financial applications. Financial Analytics User is a child application role of the Finance application role.

- **Insurance**

  Parent application role for all application roles for Insurance applications. Insurance Analytics User is a child application role of the Insurance application role.

Inheritance is used to let permissions ripple through to child application roles. The parent application roles for Financial Services and their purpose are shown in Table 2–6.

Note: A Financial Services Analytics user is provided as a child to both Finance and Insurance. Therefore, this user has permissions available to both Finance and Insurance. If you have purchased both Financial Analytics and one of the Oracle Insurance Analytics family of products (Insurance Partner Manager Analytics, Insurance Sales Analytics, Insurance Service Analytics, Insurance Marketing Analytics, Insurance Partner Manager Analytics), you should use the Financial Services Analytics user responsibilities to view all relevant dashboards.

Table 2–6 shows the additional responsibilities, and responsibility-specific security in Oracle's Financial Analytics family of products (Finance Sales Analytics, Finance Service Analytics, Finance Marketing Analytics, Finance Institutional Analytics, Finance Retail Analytics), the Oracle Insurance Analytics family of products (Insurance Partner Manager Analytics, Insurance Sales Analytics, Insurance Service Analytics, Insurance Marketing Analytics, Insurance Partner Manager Analytics), and the Oracle Healthcare Analytics family of products (Healthcare Sales Analytics, Healthcare Service Analytics, Healthcare Marketing Analytics, Healthcare Partner Manager Analytics).

If you are also deploying Usage Accelerator, Financial Services-specific Usage Accelerator responsibilities are shown in Table 2–13.

Table 2–6 Financial Services Responsibility Required to View FS Dashboards

FS Responsibilities	Dashboards
Financial Analytics User	■ Credit ■ Credit Card ■ Private Banking ■ Consumer Banking ■ Corporate and Commercial Banking ■ Investment Holdings ■ Separate Account Management ■ Wealth Management ■ Institutional Sales ■ Investment Banking ■ Finance Marketing ■ Finance Executive
User in one of the Oracle Insurance Analytics family of products (Oracle Insurance Partner Manager Analytics, Oracle Insurance Sales Analytics, Oracle Insurance Service Analytics, Oracle Insurance Marketing Analytics, Oracle Insurance Partner Manager Analytics)	■ Policy Sales ■ Policy Service ■ Insurance Marketing ■ Insurance Executive ■ Insurance Claims ■ Health Plan Sales ■ Health Plan Service ■ Health Plan Marketing ■ Health Plan Executive ■ Insurance Agents / Partners

2.8.2.4 Pharma Sales Analytics and Pharma Marketing Analytics Security Settings

Data-level security in Pharma Sales Analytics and Pharma Marketing Analytics is based on the Siebel position ID for all Pharma Analytics responsibilities except PH Executive Analytics. The Siebel position ID is always resolved through the fact table.

Data visibility is unconstrained for administrative roles. For other roles, data visibility is controlled by the position ID. The Oracle Business Analytics Warehouse uses the table W_POSITION_DH for user position-based security control. A user sees only the data that is available to that user's positions. This security model is enforced for all queries, with the exception of queries that deal exclusively with dimension data, such as:

- Time period
- Product
- Invitee status

Table 2–7 shows Pharma Analytics responsibilities and functions.

Table 2–7 Pharma Analytics Responsibilities and Functions

Responsibility	Use
LS Administrator	Administrator privileges to all options on Pharma Analytics.

Table 2–7 (Cont.) Pharma Analytics Responsibilities and Functions

Responsibility	Use
PH Call Activity Analytics Admin	Administrator privileges to Call Activity Analytics option.
PH EMEA Call Activity Analytics User	Enables brick-based metrics to be used in the Presentation Services for Pharma subject areas. Note that in the 7.7 Analytics Release, all report columns use position-based hierarchies, where in earlier releases, report columns used alignment-based sales hierarchies. All brick-based alignment pages have been removed from the reports. Therefore, if you want to use brick-based position hierarchies, you must reconfigure the reports to maintain the alternate hierarchy.
PH EMEA Executive Analytics User	Enables brick-based metrics to be used in the Presentation Services for Pharma subject areas. Note that in the 7.7 Analytics Release, all report columns use position-based hierarchies, where in earlier releases, report columns used alignment-based sales hierarchies. All brick-based alignment pages have been removed from the reports. Therefore, if you want to use brick-based position hierarchies, you must reconfigure the reports to maintain the alternate hierarchy.
PH EMEA Marketing Analytics User	Enables brick-based metrics to be used in the Presentation Services for Pharma subject areas. Note that in the 7.7 Analytics Release, all report columns use position-based hierarchies, where in earlier releases, report columns used alignment-based sales hierarchies. All brick-based alignment pages have been removed from the reports. Therefore, if you want to use brick-based position hierarchies, you must reconfigure the reports to maintain the alternate hierarchy.
PH EMEA Sales Analytics User	Enables brick-based metrics to be used in the Presentation Services for Pharma subject areas. Note that in the 7.7 Analytics Release, all report columns use position-based hierarchies, where in earlier releases, report columns used alignment-based sales hierarchies. All brick-based alignment pages have been removed from the reports. Therefore, if you want to use brick-based position hierarchies, you must reconfigure the reports to maintain the alternate hierarchy.
PH Executive Analytics Admin	Unrestricted access to all Pharma Analytics options with ZIP territories.
PH Marketing Analytics Administrator	Administrator privileges to Pharma ROI, Call Activity Profit & Loss Report, Pharma Promotional Effectiveness Subject Area, and Medical Education Effectiveness Subject Area.
PH Medical Education Analytics Admin	Administrator privileges to Medical Education Analytics option.
PH Medical Education Analytics User	Enables access to Medical Education Analytics option.
PH Sales Analytics Administrator	Administrator privileges to Rx Sales Analytics option.
PH US Call Activity Analytics User	Enables access to Call Activity Analytics Option for ZIP territory alignments.

2.8.3 About Partner Analytics Security Settings

Oracle Partner Analytics incorporates the concept of role-based analytics. Role-based analytics provides brand owners the ability to display dashboards and pages to users

based on their specific roles. For example, a sales manager would have the ability to view dashboards related to pipeline and sales effectiveness, whereas the marketing manager would have the ability to view dashboards related to campaigns. Oracle Partner Analytics also includes flexible security mechanisms to control access to subject areas and to data.

Oracle Partner Analytics roles map to Siebel responsibilities in the Siebel operational application. This section describes the roles and associated dashboards and pages for both Partner Manager and Partner Portal applications. It also includes subject area and data-level security settings for responsibilities.

2.8.3.1 PRM Partner Portal Role-Based Interactive Dashboards Mapping

The dashboard and page tab mapping for specific responsibilities in the PRM Partner Portal application are shown in Table 2–8.

Table 2–8 Responsibilities for PRM Partner Portal Analytics

Responsibility	Dashboard	Page Tab Name
Partner Executive Analytics User	Partner Executive	Pipeline, Products, Sales Effectiveness, Service
Partner Operations Analytics User	1. Partner Commerce 2. Partner Marketing 3. Partner Sales 4. Partner Service 5. Partner Training	1. Overview, Products 2. Overview, ROI 3. Pipeline, Revenue 4. Customer Sat, Overview, Service Requests 5. Training
Partner Sales Manager Analytics User	1. Partner Commerce 2. Partner Sales 3. Partner Training	1. Orders, Overview, Quotes 2. Pipeline, Revenue, Subordinates 3. Subordinates
Partner Sales Rep Analytics User	1. Partner Commerce 2. Partner Sales 3. Partner Training	1. Orders, Overview, Quotes 2. Pipeline, Revenue, Subordinates 3. Subordinates
Partner Service Manager Analytics User	1. Partner Service 2. Partner Training	1. Customer Sat, Overview, Service Requests, Subordinates 2. Subordinates
Partner Service Rep Analytics User	1. Partner Service 2. Partner Training	1. Overview, Service Requests, Subordinates 2. Subordinates

2.8.3.2 Partner Manager Role-Based Interactive Dashboards Mapping

Table 2–9 provides the dashboard and page tab mapping for specific responsibilities in the Siebel PRM Partner Manager application.

Table 2–9 Siebel Responsibilities for PRM Analytics

Responsibility	Dashboard	Page Tab Name
Channel Account Manager Analytics User	1. Channel Customers 2. Channel Sales 3. Channel Service 4. Channel Training	1. Overview, Sales 2. Products, Sales 3. Products, Service 4. Training Profile
Channel Executive Analytics User	1. Channel Customers 2. Channel Executive 3. Channel Segmentation	1. Customer Profile 2. Customer Satisfaction, Pipeline, Product, Program, Revenue, Service 3. Channel Mix, Partner Territory, Partner Tier, Partner Type
Channel Marketing Manager Analytics User	1. Channel Customers 2. Customer Marketing	1. Overview, Sales 2. Effectiveness, Responses, ROI
Channel Operations Analytics User	1. Channel Commerce 2. Channel Customers 3. Channel Marketing 4. Channel Sales 5. Channel Segmentation 6. Channel Service 7. Channel Training	1. Orders, Overview, Quotes, Products 2. Overview, Sales, Service 3. Effectiveness, Overview 4. Margins, Pipeline, Revenue, Sales Cycle, Wins 5. Partner Territory, Partner Tier, Partner Type 6. Customer Satisfaction, Overview, Products, Resolution Time, Service Requests 7. Overview, Performance

2.8.3.3 PRM Analytics Subject Area Mappings

Ad hoc queries in Siebel PRM Analytics are built by the user, depending on user responsibilities and based on columns in subject areas in the Oracle BI application. By restricting visibility to subject areas based on responsibilities, PRM Analytics provides brand owners a flexible way to deploy role-based analytics.

The subject area visibility for responsibilities in Partner Manager are shown in Table 2–10, where an X indicates that the subject area is visible for the user holding that responsibility.

Table 2–10 Responsibilities for PRM Partner Manager Analytics

Subject Area	Channel Executive Analytics User	Channel Operations Analytics User	Channel Account Manager Analytics User	Channel Marketing Manager Analytics User
Activities	X	X	X	X
Assets	X	X	X	-
Campaigns	X	X	X	X
Consumers	X	X	X	X
Customer Satisfaction	X	X	X	-

Table 2–10 (Cont.) Responsibilities for PRM Partner Manager Analytics

Subject Area	Channel Executive Analytics User	Channel Operations Analytics User	Channel Account Manager Analytics User	Channel Marketing Manager Analytics User
Customers	X	X	X	X
Orders	X	X	X	X
Partner Training	X	X	X	-
Partners	X	X	X	X
Pipeline	X	X	X	X
Pricing	X	X	X	X
Products	X	X	X	X
Real-Time Activity	-	-	-	-
Real-Time Assets	-	-	-	-
Service Requests	X	X	X	-

2.8.3.4 PRM Analytics Subject Area Visibility

The subject area visibility for roles in Partner Portal is shown in Table 2–11, where an X indicates that subject area is visible for the user holding that responsibility.

Table 2–11 Subject Area Visibility for PRM Partner Portal

Subject Area	Partner Executive Analytics User	Partner Operations Manager Analytics User	Partner Sales Manager Analytics User	Partner Sales Rep Analytics User	Partner Service Manager Analytics User	Partner Service Rep Analytics User
Activities	X	X	X	X	X	X
Assets	X	X	-	-	X	X
Campaigns	X	X	-	-	-	-
Consumers	X	X	-	-	-	-
Customer Satisfaction	X	X	-	-	X	X
Customers	X	X	X	X	X	X
Orders	X	X	X	X	X	X
Partner Training	X	X	X	X	X	X
Partners	X	X	-	-	-	-
Pipeline	X	X	X	X	-	-
Pricing	-	-	-	-	-	-
Products	X	X	X	X	X	X
Real-Time Activity	-	-	-	-	-	-
Real-Time Assets	-	-	-	-	-	-

Table 2–11 (Cont.) Subject Area Visibility for PRM Partner Portal

Subject Area	Partner Executive Analytics User	Partner Operations Manager Analytics User	Partner Sales Manager Analytics User	Partner Sales Rep Analytics User	Partner Service Manager Analytics User	Partner Service Rep Analytics User
Service Requests	X	X	-	-	X	X

2.8.3.5 PRM Analytics Data-Level Visibility

PRM Analytics also provides brand owners the ability to restrict security based on the user's organization or position. This security mechanism makes sure that one user does not have access to another user's data. It also makes sure that one partner does not have access to another partner's data. Data-level security is administered for responsibilities. Details regarding setting up data -level visibility are provided in the topic Section 2.2.2, "Implementing Data-Level Security in the Oracle BI Repository."

Table 2–12 shows the data-level security settings included for the responsibilities in Partner Manager and Partner Portal.

Table 2–12 Oracle PRM Data-Level Security Settings

Responsibility	Data-Level Security	Type	Comments
Channel Executive Analytics User	No	N/A	N/A
Channel Operations Analytics User	No	N/A	N/A
Channel Account Manager Analytics User	No	N/A	N/A
Channel Marketing Manager Analytics User	No	N/A	N/A
Partner Executive Analytics User	Yes	Organization	Displayed records should match organization of the user.
Partner Sales Manager Analytics User	Yes	Organization	Displayed records should match organization of the user.
Partner Sales Rep Analytics User	Yes	Position	Displayed records should match position of the user.
Partner Service Manager Analytics User	Yes	Organization	Displayed records should match organization of the user.
Partner Service Rep Analytics User	Yes	Position	Displayed records should match position of the user.

2.8.4 About Usage Accelerator Analytics Security Settings

Table 2–13 describes the additional security configurations that may be necessary and the particular responsibilities associated with the Oracle Usage Accelerator dashboards.

Table 2–13 Usage Accelerator Responsibilities and Dashboards

User Responsibility	Data-Level Security	Dashboard Name (View)	Dashboard Page
Usage Accelerator–Sales Rep	Primary Position Data-Level Security	Score Card	Individual Scorecard
Usage Accelerator–Financial Services Sales Rep	-	Action Plan	Account Coverage Contact Coverage Opportunity Coverage Financial Account Coverage—Financial Services only Account Completeness Contact Completeness Opportunity Updates
Usage Accelerator–Sales Manager	No position-based Security	1. Score Card 2. Action Plan 3. Master Data Management - Customer Hub	1. Team Scorecard, Individual Scorecard 2. Account Coverage (Team), Contact Coverage (Team), Opportunity Coverage (Team), Financial Account Coverage (Team)—Financial Services only, Account Completeness (Team), Contact Completeness (Team), Opportunity Updates (Team) 3. Master Record Completeness, Master Record Completeness Detail, Accuracy
Usage Accelerator–Financial Services Sales Manager	-	1. Coverage 2. Completeness 3. Opportunity Updates 4. User Adoption	1. Account Coverage, Account Coverage (Team), Contact Coverage, Opportunity Coverage, Financial Account Coverage—Financial Services only 2. Account Completeness, Contact Completeness 3. Opportunity Updates 4. Active Users, Application Usage—excluded for Financial Services, Application Usage—Financial Services only*
Usage Accelerator–Sales Executive	No position-based Security	1. Scorecard 2. Master Data Management - Customer Hub	1. Organization Scorecard, Individual Scorecard 2. Master Record Completeness, Master Record Completeness Detail, Accuracy
Usage Accelerator–Financial Services Sales Executive	-	Action Plan	Account Coverage (Org) Contact Coverage (Org) Opportunity Coverage (Org) Financial Account Coverage (Org)—Financial Services only Account Completeness (Org) Contact Completeness (Org) Opportunity Updates (Org)

2.8.5 About Primary Owner-Based Security

Primary owner-based security is supported through the "Primary Owner-Based Security" application role. This type of security mechanism allows records to be visible only to their primary owner. By default, this type of security supports a few dimensions in the Core business model, but other tables can be added if they have a primary owner's source Integration ID column.

The security filter in this application role is defined as:

"Core"."Dim - Activity"."VIS_PR_OWN_ID" = VALUEOF(NQ_SESSION."PR_OWNER_ID")

The session variable PR_OWNER_ID is a single value variable, populated by the Primary Owner ID initialization block. This initialization block runs the following SQL, for the Siebel OLTP data source, to populate the variable:

```
select PAR_ROW_ID

from VALUEOF(TBO).S_USER

where LOGIN = 'VALUEOF(NQ_SESSION.USER)'
```

2.8.6 About Business Unit-Based Security

Business unit-based security is supported through the "Primary Org-Based Security" application role. By default, only a few dimensions in the Core, Workforce Analytics and Forecasting business models support this data security type. Other fact and dimension tables can be added to this application role if they have the column VIS_PR_BU_ID column populated.

The security filter in this application role is defined as:

```
"Core"."Dim - Order"."VIS_PR_BU_ID" = VALUEOF(NQ_
SESSION."ORGANIZATION")
```

The session variable ORGANIZATION is a Row-wise variable, initialized using the Initialization block: Orgs for Org-Based Security. This Init Block runs the following SQL for the Siebel OLTP data source, to populate the ORGANIZATION variable:

```
select distinct 'ORGANIZATION', PRR.SUB_PARTY_ID

from VALUEOF(TBO).S_POSTN P, VALUEOF(TBO).S_USER U,
VALUEOF(TBO).S_PARTY_PER PP, VALUEOF(TBO).S_PARTY_RPT_REL PRR

where U.ROW_ID=PP.PERSON_ID and P.ROW_ID=PP.PARTY_ID and
PRR.PARTY_ID = P.BU_ID and PRR.PARTY_TYPE_CD = 'Organization'
and U.LOGIN = 'VALUEOF(NQ_SESSION.USER)'
```

2.9 About Security Integration with Oracle's JD Edwards EnterpriseOne or JD Edwards World

All information in this section pertaining to JD Edwards EnterpriseOne also applies to JD Edwards World.

This section covers an approach to security integration between Oracle Business Intelligence Enterprise Edition (Oracle BI EE) and JD Edwards EnterpriseOne using the Lightweight Directory Access Protocol (LDAP). It contains the following topics:

- Section 2.9.1, "How Oracle BI EE and JD Edwards EnterpriseOne Use LDAP"
- Section 2.9.2, "Integration of User and Object Security"

- Section 2.9.3, "Implementing LDAP Integration for User and Object Security"

2.9.1 How Oracle BI EE and JD Edwards EnterpriseOne Use LDAP

LDAP can serve as a central repository of security information for both JD Edwards EnterpriseOne and Oracle BI EE, allowing administrators to configure security once for both systems. An LDAP server stores credentials required for authentication, as well as user profile information such as JD Edwards EnterpriseOne roles and user groups.

At login, the Oracle BI EE server passes a user's credentials to the LDAP server for authentication. Upon successful authentication, an Oracle BI EE initialization block retrieves the group names from the user's LDAP record. These group names are stored in the Oracle BI EE session variable GROUP and are then matched with the list of available application roles in the policy store to grant the appropriate permissions to the user. This information is used throughout the user's session to determine which applications, dashboards, and other objects the user has permission to access.

Similarly, at login the JD Edwards EnterpriseOne security kernel passes a user's credentials to the LDAP server. Upon successful authentication, the JD Edwards EnterpriseOne security kernel retrieves the user-role relationship information which is used for both object and data security.

2.9.2 Integration of User and Object Security

LDAP can provide an integration for Oracle Business Intelligence Enterprise Edition and JD Edwards EnterpriseOne for user and object security only. LDAP cannot provide an integrated data security solution. Therefore, to implement data security, you must configure security separately on each server. This requires user authentication to be set up on both the Oracle Business Intelligence Enterprise Edition server and the JD Edwards EnterpriseOne server.

2.9.3 Implementing LDAP Integration for User and Object Security

This section contains the following topics:

- Section 2.9.3.1, "About Configuring Oracle Business Intelligence Enterprise Edition to Use LDAP"
- Section 2.9.3.2, "About Configuring JD Edwards EnterpriseOne to Use LDAP"

2.9.3.1 About Configuring Oracle Business Intelligence Enterprise Edition to Use LDAP

For instructions on how to configure Oracle Business Intelligence Enterprise Edition to allow authentication of users through LDAP, see the section about setting up alternative authentication providers in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

2.9.3.2 About Configuring JD Edwards EnterpriseOne to Use LDAP

For instructions on how to configure JD Edwards EnterpriseOne to allow authentication of users through LDAP, see the *JD Edwards EnterpriseOne Tools Security Administration Guide*.

2.9.3.3 About Configuring JD Edwards World to Use LDAP

For instructions on how to configure JD Edwards World to allow authentication of users through LDAP, see the *JD Edwards World Technical Foundation Guide*.

Index

L

M

O

P

R

S

T

U