Oracle BI 11*g* R1: Build Repositories

Volume II - Student Guide

D63514GC10

Edition 1.0

October 2010

D69304

Author

Jim Sarokin

Technical Contributors and Reviewers

Marla Azriel
Roger Bolsius
Bob Ertl
Alan Lee
Monica Moore
Kasturi Shekhar
Aneel Shenker
Scott Silbernick
Nick Tuson
Krishnan Viswanathan

Graphic Designer

Rajiv Chandrabhanu

Editors

Aju Kumar
Daniel Milne
Richard Wallis

Publishers

Syed Ali
Giri Venugopal

Contents

3 Building the Physical Layer of a Repository

4 Building the Business Model and Mapping Layer of a Repository

19

Security

Objectives

After completing this lesson, you should be able to:

- Identify and describe default security settings for Oracle BI
- Create users and groups
- Create application roles
- Set up permissions for repository objects
- Use query limits, timing restrictions, and filters to control access to repository information

Business Challenge: Security Strategy

Security strategy designs for a business start with answers to these basic questions:

- Who will have access to company data and business resources?
- Under what conditions will access be limited or denied?
- How will access be enforced?
- How will users authenticate themselves?
- Where will credentials be stored?

Business Challenge: Security Strategy

An organization's business needs guide its security strategy. Every organization has unique information-protection requirements, and each security strategy must be individually designed to match. Many security strategy designs start with answers to the questions listed in the slide.

Business Solution: Oracle BI Security

Controls access to system resources:

- Requires users to authenticate at login
- Restricts users to only those resources for which they are authorized
- Manages user identities, credentials, and permission grants

Business Solution: Oracle BI Security

Securing Oracle Business Intelligence can be divided into two broad areas: controlling access to the components within the BI domain (resource access security) and controlling access to business source data (data access security).

Controlling access to system resources is achieved by requiring users to authenticate at login and by restricting users to only those resources for which they are authorized. The Oracle BI default security model is available for immediate implementation after installation. This model includes controls for managing user identities, credentials, and permission grants. This allows you to control system access by validating identity at login (authentication) and control access to specific Oracle BI components and features according to a user's permission grants (authorization).

Note: This course highlights the security components that relate to building and managing an Oracle BI repository. However, Oracle BI Security is a complex subject that involves installation and configuration tasks that are beyond the scope of this course. For more detailed information, please refer to the *Security Guide for Oracle Business Intelligence Enterprise Edition*.

Managing Oracle BI Security

Oracle BI integrates with Oracle Fusion Middleware's security platform.

- Oracle WebLogic Server Administration Console
  - Management of users and groups for the embedded LDAP server that serves as the out-of-the-box default identity store
- Oracle Enterprise Manager Fusion Middleware Control
  - Management of policy store application roles that grant permissions to users, groups, and other application roles
- Oracle BI Administration Tool
  - Management of permissions for Presentation layer objects and business model objects in the repository

Managing Oracle BI Security

Oracle BI software integrates seamlessly with the Oracle Fusion Middleware platform. They share a common security framework and features. This common security configuration uses Oracle WebLogic Server as the default administration server. However, these implementation details are largely hidden while performing daily administrative tasks and are exposed only by the tools used to manage your Oracle BI security configuration.

Oracle BI Default Security Model

During installation, three Oracle BI security controls are preconfigured with initial (default) values to form the default security model:

- Identity store
  - Contains the definitions of users, groups, and group hierarchies required to control authentication
- Policy store
  - Contains the definition of application roles, the permissions granted to the roles, and the members (users, groups, and applications roles) of the roles
- Credential store
  - Stores security-related credentials, such as user name and password combinations, for accessing an external system (such as a database or LDAP server)

Oracle BI Default Security Model

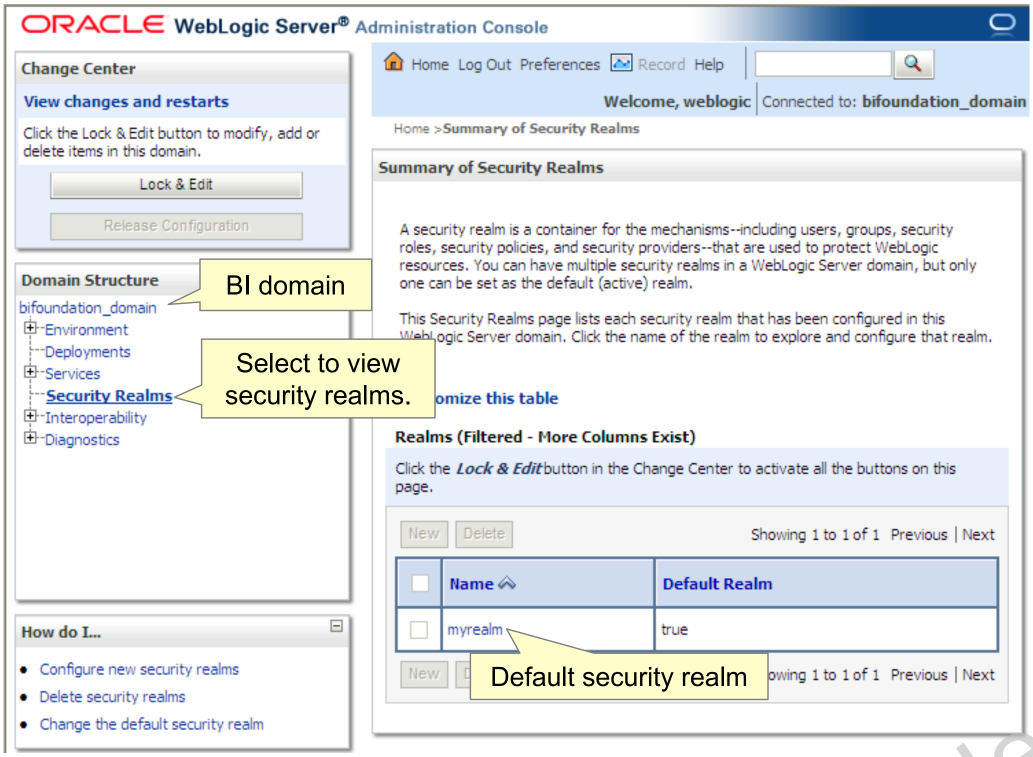
The security controls include:

- An embedded directory server functioning as an **identity store** designed to hold all user and group definitions that are required to control authentication
- A file-based **policy store** designed to hold the application-role and permission-grant mappings to users and groups that are required to control authorization
- A file-based **credential store** designed to hold user and system credentials

When operating in a development or test environment, you may find it convenient to use the default security model because it comes preconfigured. You then add user definitions and credentials that are specific to your business as well as customize the default application roles and permission grants that your business security policies require. After the identity, policy, and credential stores are fully configured and populated with data that is specific to your business, they provide all user, policy, and credential information needed by the Oracle BI components during authentication and authorization.

The following slides cover the key components of the default security model.

Default Security Realm



Default Security Realm

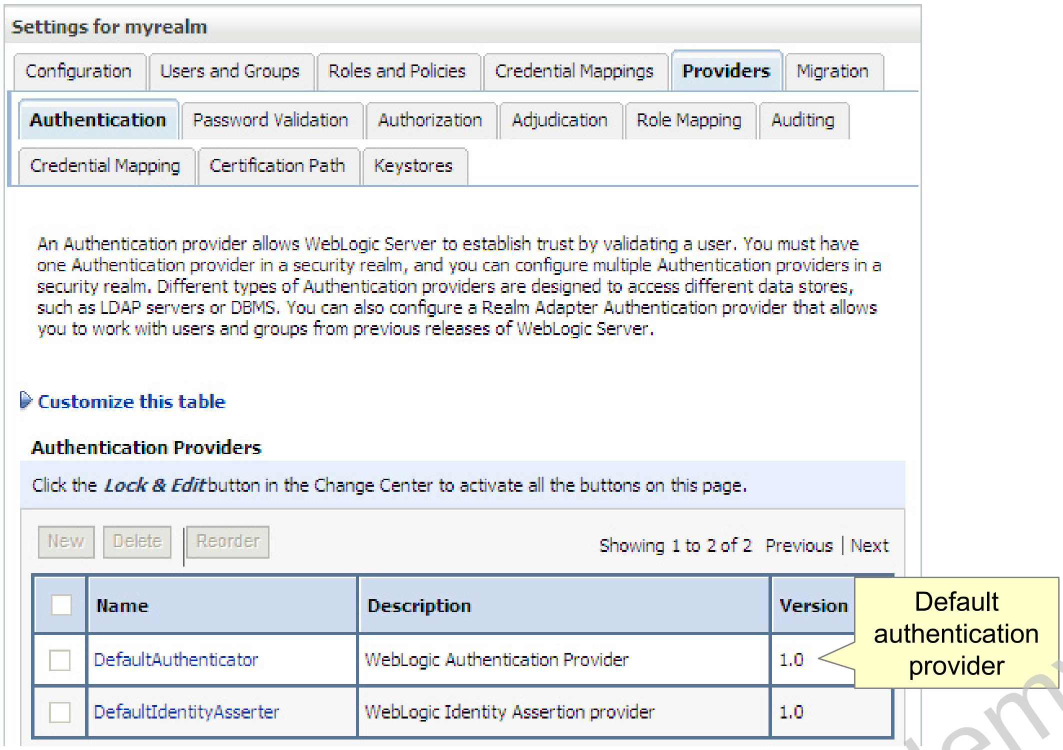
The screenshot shows the default setting for the Oracle BI WebLogic security realm in the Oracle WebLogic Server Administration Console. Using a browser, you can access the Administration Console with the following URL: http://<machine name>:7001/console.

On the left side of the console, under Domain Structure, notice that there is a single WebLogic domain named `bifoundation_domain` into which all of the BI applications are deployed. Notice also that there is a single default security realm named `myrealm`. The OBI installer installs a single domain with a single security realm in it.

A security realm is a container for the mechanisms that are used to protect WebLogic resources. This includes users, groups, security roles, security policies, and security providers. Whereas multiple security realms can be defined for the BI domain, only one can be active (that is, only one can be designated as the default realm at any given time).

Click `myrealm` to view its default settings.

Default Authentication Providers



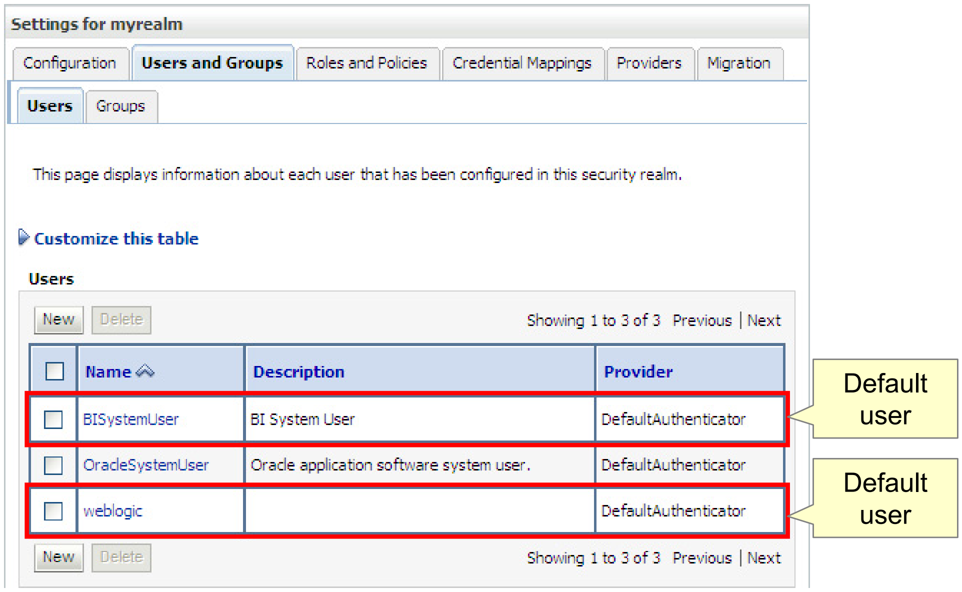
Default Authentication Providers

This screenshot shows the default settings for authentication providers in the default `myrealm` security realm. Note that there is a default WebLogic Authentication Provider. An authentication provider establishes the identity of users and system processes, transmits identity information, and serves as a repository from which components can retrieve identity information.

Oracle BI is configured out of the box to use the directory server embedded in Oracle WebLogic Server as the default security provider. When a user logs in to a system with a user name and password combination, Oracle WebLogic Server validates identity based on the combination provided. Alternative security providers can be used if desired and managed in the Oracle WebLogic Administration Console, but the WebLogic Authentication Provider is used by default.

Note: There is a default WebLogic Identity Assertion Provider, which is used primarily for Single Sign On and is not covered in this training.

Default Users



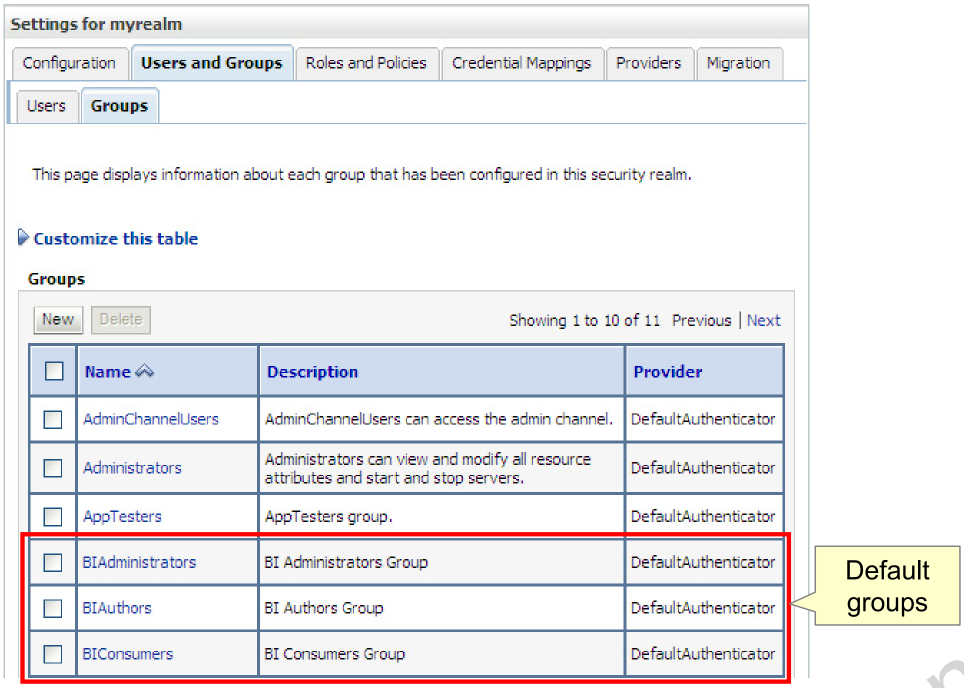
Default Users

The default identity store contains user names that are specific to Oracle BI. These default user names are provided as a convenience so you can begin using the Oracle BI software immediately after installation, but you are not required to maintain the default names. In this example, the users are `BISystemUser` and `weblogic`.

`weblogic` is the administrative user. After installation, a single administrative user is shared by Oracle BI and Oracle WebLogic Server. The same user name and password that were supplied during the installation process are used for both. The user name that is created during installation can be any desired name and does not need to be Administrator. The password is also provided during installation and afterward can be changed by using the administration interface for the identity store. In the default security configuration, an administrative user is a member of the BIAdministrators group and has all rights granted to the Oracle BI Administrator user in earlier releases, with the exception of impersonation. The administrative user cannot impersonate other users.

Oracle BI system components now establish a connection to each other as `BISystemUser` instead of as the Administrator (the latter being the practice in earlier releases). Using a trusted system account such as `BISystemUser` to secure communication between components enables you to change the password of your deployment's system administrator account without affecting communication between these components. The name of this user is the default and can be changed, or a different user can be created for the purpose of interprocess communication. This is a highly privileged user whose credentials should be protected from nonadministrative users.

Default Groups



Name	Description	Provider
AdminChannelUsers	AdminChannelUsers can access the admin channel.	DefaultAuthenticator
Administrators	Administrators can view and modify all resource attributes and start and stop servers.	DefaultAuthenticator
AppTesters	AppTesters group.	DefaultAuthenticator
BIAdministrators	BI Administrators Group	DefaultAuthenticator
BIAuthors	BI Authors Group	DefaultAuthenticator
BIConsumers	BI Consumers Group	DefaultAuthenticator

Default Groups

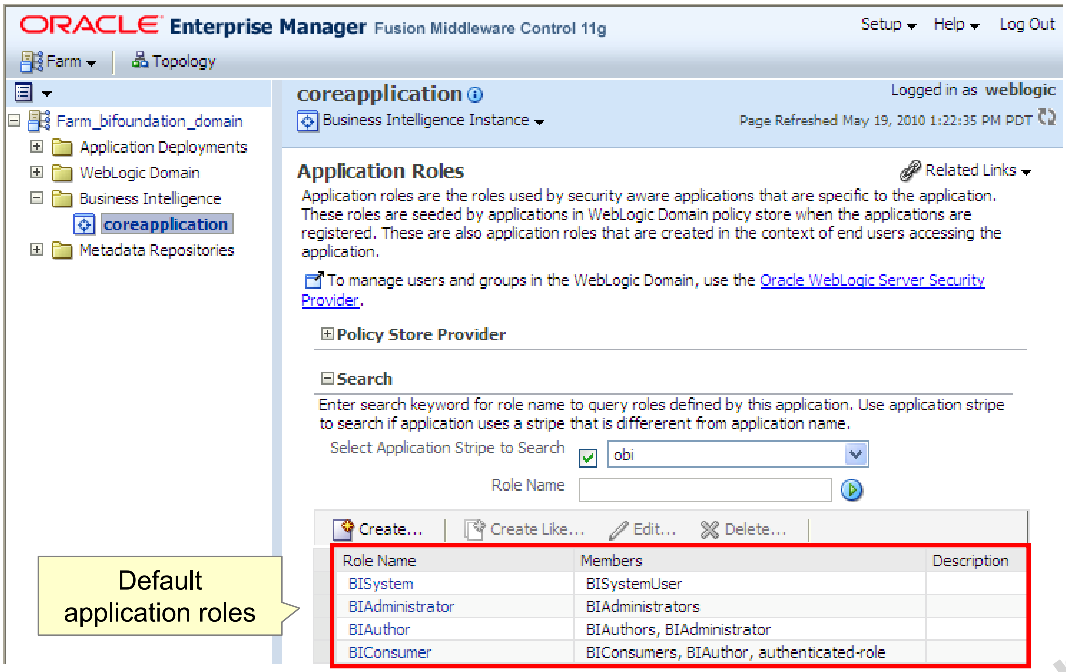
Groups are logically ordered sets of users. Creating groups of users who have similar needs for access to system resources enables easier security management. Managing a group is more efficient than managing a large number of users individually. Oracle recommends that you organize your users into groups for easier maintenance. Groups are then mapped to application roles in order to grant rights. Three default group names are provided as a convenience so you can begin using the Oracle BI software immediately after installation, but you are not required to maintain the default names.

BIAdministrators group: Members have the equivalent permissions of the Administrator user of earlier releases with the exception of the ability to impersonate. The Administrator user of earlier releases could impersonate, but members of the BIAdministrators group cannot impersonate other users.

BIAuthors group: Members have the permissions necessary to create content for other users to use (or to consume).

BIConsumers group: Members have the permissions necessary to use (or consume) content created by other users. The BIConsumers group represents all users who have been authenticated by Oracle BI. By default, every Oracle BI authenticated user is part of the BIConsumers group and does not need to be explicitly added to the group. The BIConsumers group includes the Oracle WebLogic Server users group as a member.

Default Application Roles



Default Application Roles

An application role defines a set of permissions that are granted to a user or group. Application roles are defined in Fusion Middleware Control Enterprise Manager, which can be accessed via http://<machine name>:7001/em. To access the Application Roles page, right-click `coreapplication` in the left pane and select Security > Application Roles (not shown here).

The file-based policy store contains a default application role hierarchy that includes preconfigured permissions grants and role membership definitions. Application role members can include users or groups from the identity store or other application roles from the policy store.

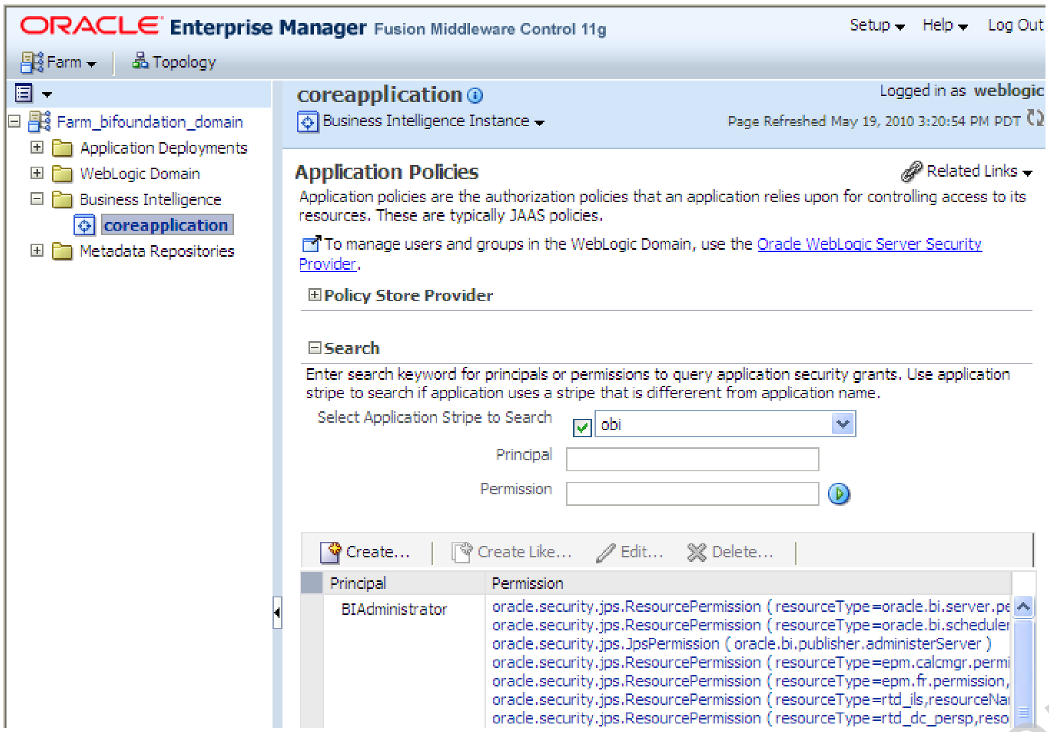
Default application roles include:

- **BISystem:** Grants the permissions necessary to impersonate other users. This role is required by Oracle BI system components for intercomponent communication.
- **BIAdministrator:** Grants the administrative permissions necessary to configure and manage the Oracle BI installation. Any member of the BIAdministrators group is explicitly granted this role and implicitly granted the BIAuthor and BIConsumer roles.
- **BIAuthor:** Grants the permissions necessary to create and edit content for other users to use (or to consume). Any member of the BIAuthors group is explicitly granted this role and implicitly granted the BIConsumer role.
- **BIConsumer:** Grants the permissions necessary to use (or to consume) content created by other users

Default Application Roles (continued)

These default application roles map to default groups in the default WebLogic LDAP. The groups are listed in the Members column. Application roles are independent of LDAP groups. If you moved to a different LDAP server rather than the default WebLogic LDAP server, you could map these roles to groups in the new LDAP server. Application roles are in the policy store, whereas groups are in the identity store. The default naming convention is that application role names are singular and group names are plural.

Default Application Policies



Default Application Policies

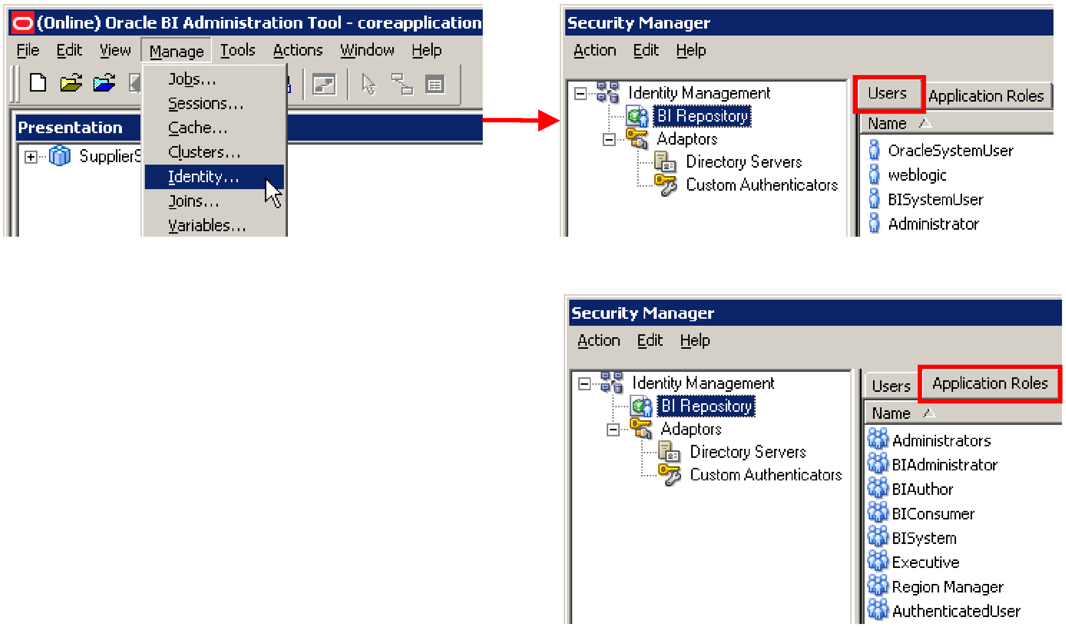
Application policies are the authorization policies that an application relies upon for controlling access to its resources. Application policies are defined in Fusion Middleware Control Enterprise Manager. To access the Application Policies page, right-click `coreapplication` in the left pane and select Security > Application Policies (not shown here).

The default file-based policy store contains the Oracle BI permissions. All Oracle BI permissions are provided; you cannot create additional permissions. These permissions are granted by the default application roles in the default security configuration. Each default application role has a predefined set of permissions. The screenshot shows only a partial list of permissions for the BIAdministrator application role.

An example of a permission is `oracle.bi.server.manageRepositories`, which grants permission to open repositories in online mode in the Oracle BI Administration Tool. This permission is granted to the BIAdministrator role.

Note: These policy store permissions are not the same as those used to define access to BI objects (metadata, dashboards, reports, and so on). Policy store permissions are used only to define the BI functionality that assigned roles can access. You learn more about object permissions later in this lesson.

Default Security Settings in the Repository



Default Security Settings in the Repository

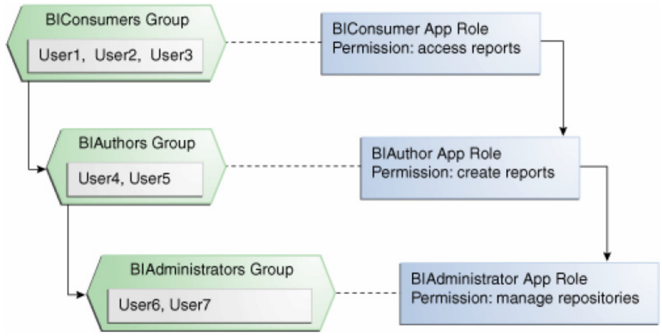
Open the repository in online mode to see the default security settings. Repository security should be managed in online mode. Select Manage > Identity to open the Security Manager.

Notice that on the Users tab you can see the same set of users as those listed in the WebLogic Server Administration Console. The key point is that users are no longer managed in the repository, as they were in prior releases. They are managed in the WebLogic LDAP, or whatever identify store your system is configured with. If you need to add a new user to the system, you must do it in the identity store, not the repository (as you learn later in this lesson).

The Application Roles tab shows all application roles in the policy store. Later in this lesson you learn that you can use the application roles to set access control permissions for repository objects. The recommended practice is to use application roles, not individual users, to set access control permissions for repository objects.

If you create new users in the identity store, or new application roles in the policy store, they appear in the repository after BI Server is restarted. The repository holds a cache of the identities, so users and application roles are visible in offline mode as well as online mode.

Default Application Role Hierarchy: Example



User Name	Group Membership: Explicit/Inherited	Application Role Membership: Explicit/Inherited	Permission Grants: Explicit/Inherited
User1 User2 User3	BIConsumers: Explicit	BIConsumer: Explicit	Access reports: Explicit
User4 User5	BIAuthors: Explicit BIConsumers: Inherited	BIAuthor: Explicit BIConsumer: Inherited	Develop reports: Explicit Access reports: Inherited
User6 User7	BIAdministrators: Explicit BIAuthors: Inherited BIConsumers: Inherited	BIAdministrator: Explicit BIAuthor: Inherited BIConsumer: Inherited	Manage repository: Explicit Develop reports: Inherited Access Reports: Inherited

ORACLE

Default Application Role Hierarchy: Example

This example is provided to illustrate the relationships among users, groups, application roles, and permissions. In Oracle BI, the members of a default application role include both groups and other application roles. The result is a hierarchical role structure in which permissions can be inherited in addition to being explicitly granted. A group that is a member of a role is granted both the permissions of the role and the permissions for all roles descended from that role. When you construct a role hierarchy, you should *not* introduce circular dependencies.

The graphic in the slide shows these relationship among the default application roles and the ways in which permissions are granted to members. The result is that, because of the role hierarchy, a user who is a member of a particular group is granted both explicit permissions and any additional inherited permissions. Note that, by themselves, groups and group hierarchies do not allow a privilege to perform an action in an application. Those privileges are conveyed through the application roles and their corresponding permission grants.

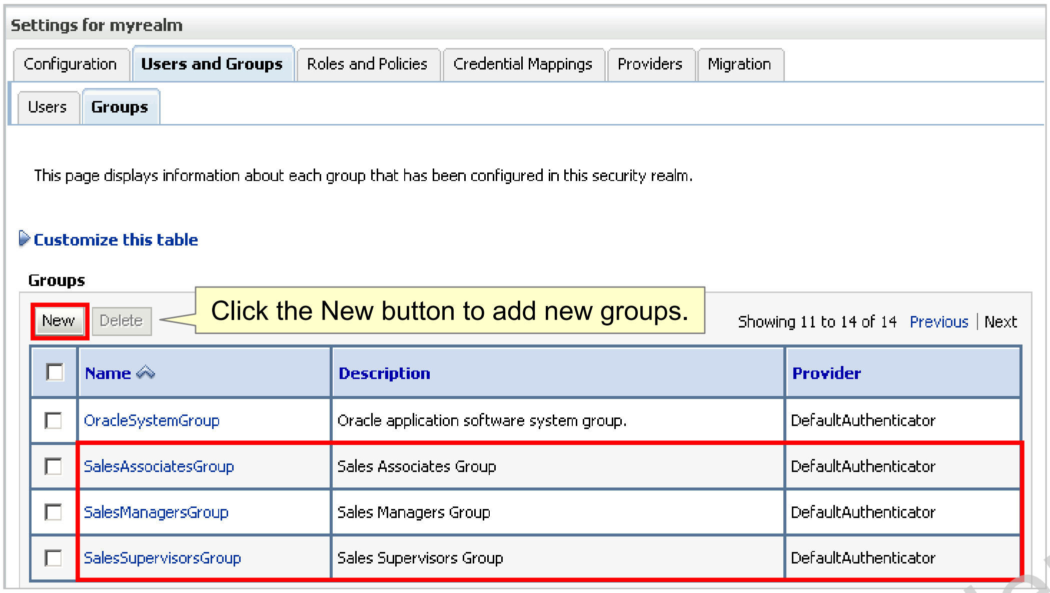
The table shows the role and permissions granted to all group members (users). The default BIAdministrator role is a member the BIAuthor role, and BIAuthor role is a member of the BIConsumer role. The result is that members of the BIAdministrators group are granted all the permissions of the BIAdministrator role, the BIAuthor role, and the BIConsumer role. In this example, only one of the permissions granted by each role is used for demonstration purposes.

ABC Example

1. Create groups.
2. Create group hierarchies.
3. Create users.
4. Assign users to groups.
5. Create application roles.
6. Assign groups and roles to application roles.
7. Verify new users and application roles in Oracle BI.
8. Set up object permissions.
9. Set row-level security (data filters).
10. Set query limits and timing restrictions.

Create Groups

Use the security realm in the WebLogic Server Administration Console to create groups.



Create Groups

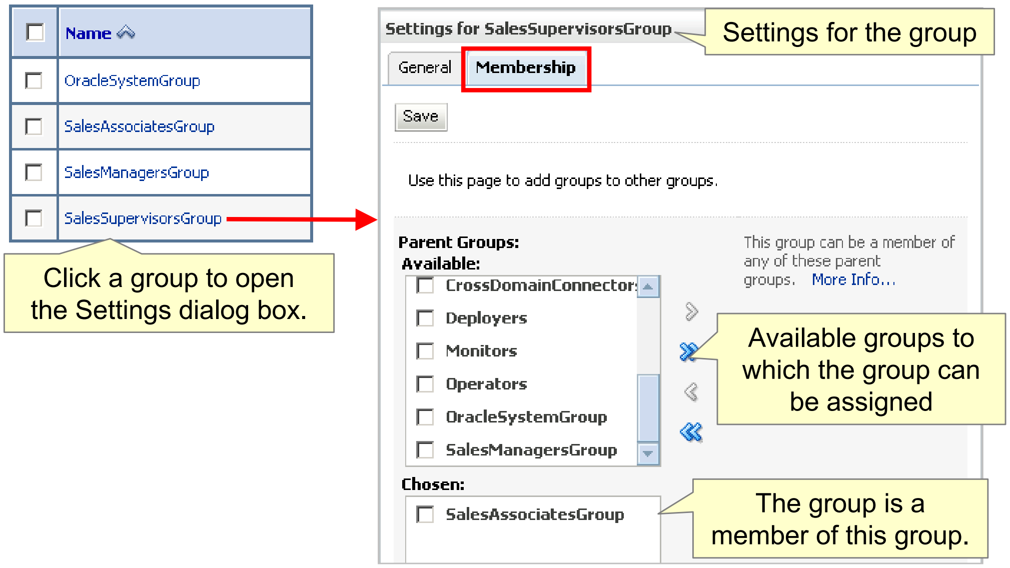
You use the WebLogic Server Administration Console to create groups. Groups are logical ordered sets of users. Managing a group is more efficient than managing a large number of users individually. Oracle recommends that you first organize all Oracle BI users into groups that make sense for your organization's goals and map application roles to the groups to convey system privileges. The out-of-the-box identity store provided for managing users and groups is Oracle WebLogic Server's embedded directory server.

In this example, three new groups are added: SalesAssociatesGroup, SalesManagersGroup, and SalesSupervisorsGroup.

When you click the New button, a dialog box opens to create a new group. The dialog box is not shown here.

Create Group Hierarchies

Add groups to other groups to create group hierarchies.



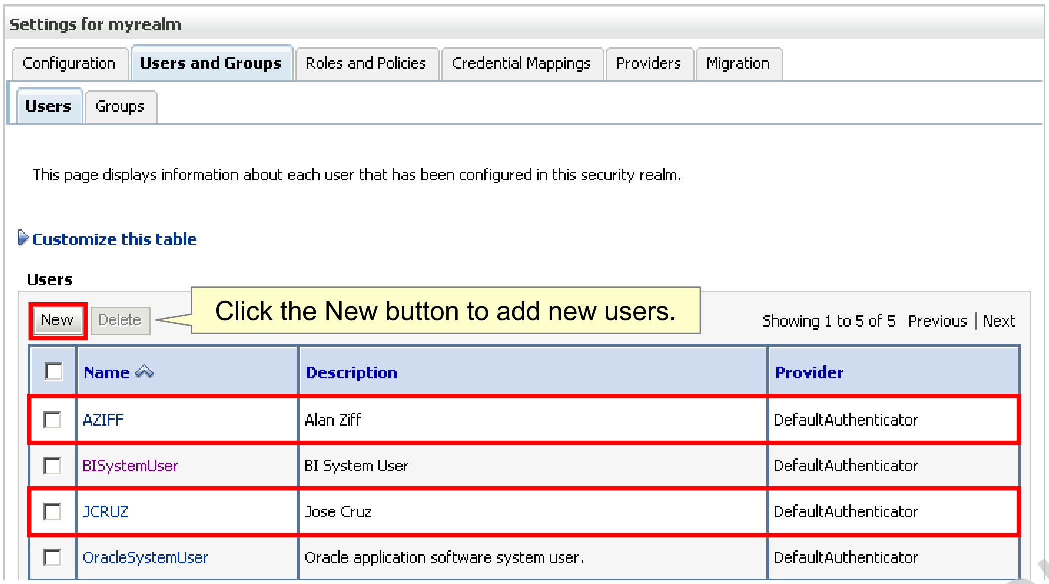
Create Group Hierarchies

You use the security realm in the WebLogic Server Administration Console to create group hierarchies. On the “Users and Groups” tab in the security realm, click a group on the Groups subtab to view settings for the group. On the Membership subtab, you can assign groups to other groups. The Membership subtab shows the groups of which a group is already a member and available groups to which a group can be assigned. To assign a group, use the buttons to move a group (or groups) from the Available list to the Chosen list.

The example in the slide shows the group membership settings for the SalesSupervisorsGroup group. The SalesSupervisorsGroup group is a member of the SalesAssociatesGroup group. This means that any privileges or permissions assigned to the Sales Associates group are inherited by the Sales Supervisors group.

Create Users

Use the security realm in the WebLogic Server Administration Console to create users.



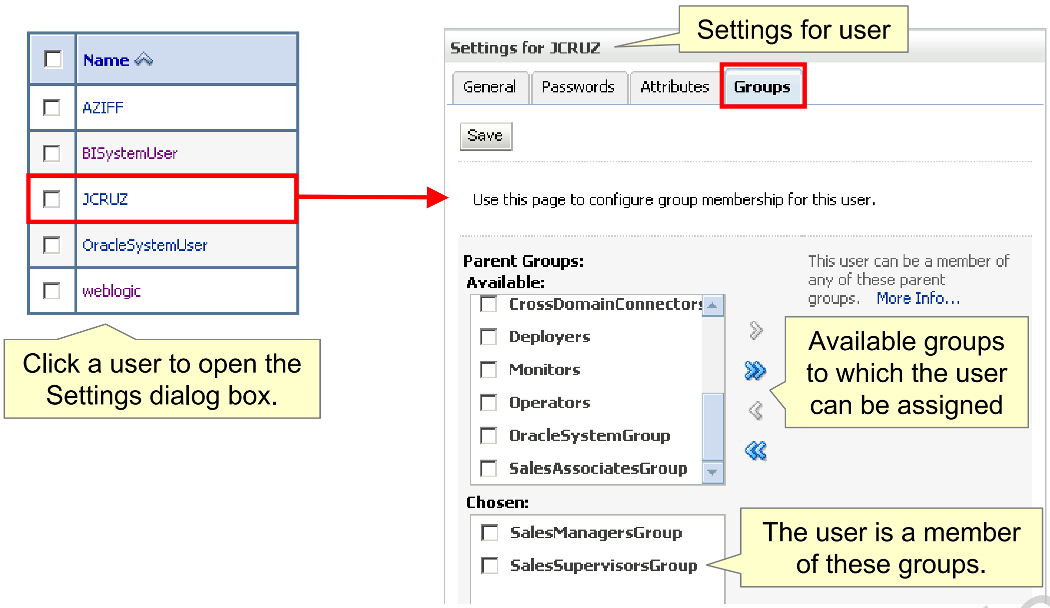
Create Users

You use the WebLogic Server Administration Console to create users. The out-of-the-box identity store provided for managing users is Oracle WebLogic Server's embedded directory server. In this example, two new users are added: `AZIFF` and `JCRUZ`.

When you click the New button, a dialog box opens to create a new user. In the dialog box, you provide the user name, description, and password. (The dialog box is not shown here.)

Assign Users to Groups

Use the security realm in the WebLogic Server Administration Console to assign users to groups.



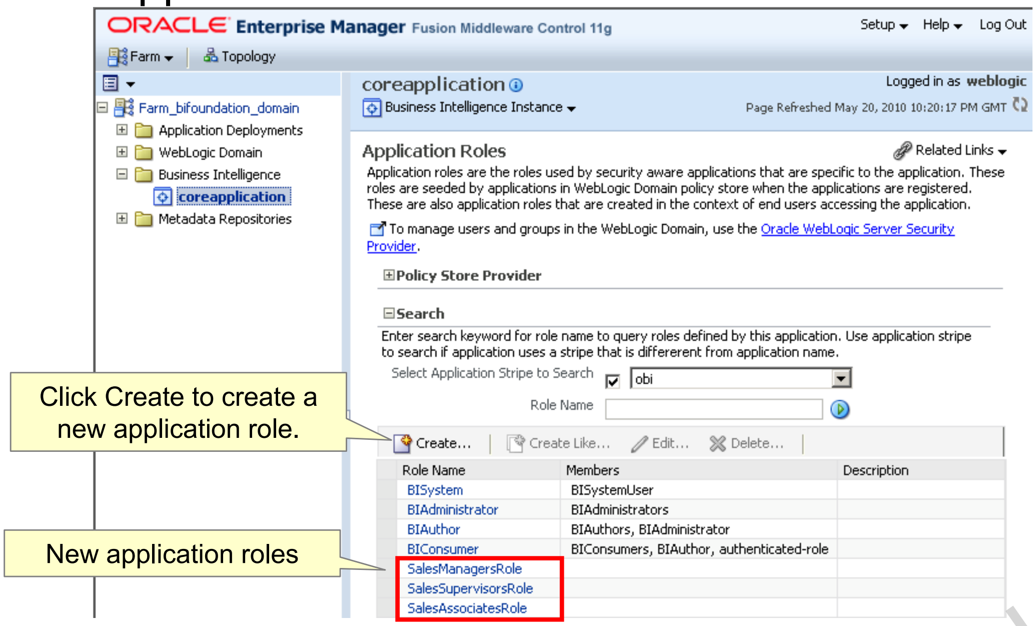
Assign Users to Groups

On the "Users and Groups" tab in the security realm, click a user on the Users subtab to view settings for the user. On the Groups subtab, you can assign users to groups. The Groups subtab shows the groups of which a user is already a member and available groups to which a user can be assigned.

The example in the slide shows the group settings for the `JCRUZ` user. `JCRUZ` is a member of the Sales Managers and Sales Supervisors groups.

Create Application Roles

Use Fusion Middleware Control Enterprise Manager to create application roles.



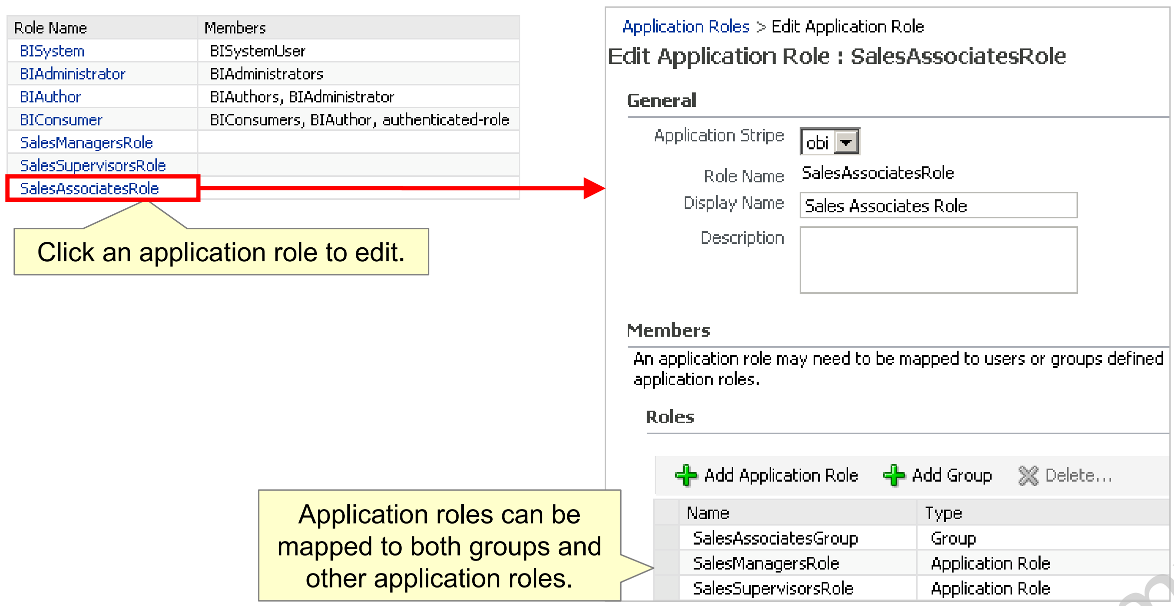
Create Application Roles

An application role conveys its permission grants to the users, groups, and application roles that are mapped to that role. Being mapped to an application role establishes membership in the role. Binding the permission grants to the application role streamlines the process of granting system privileges. Once the application role and permission grant definitions are established, you control system rights by managing membership in each role.

Oracle recommends that you map groups and other application roles to application roles and not to individual users. Once mapped, all members of the groups and roles are granted the same rights. Controlling membership in a group reduces the complexity of tracking access rights for multiple individual users.

Map Application Roles

Map application roles to groups or other application roles.



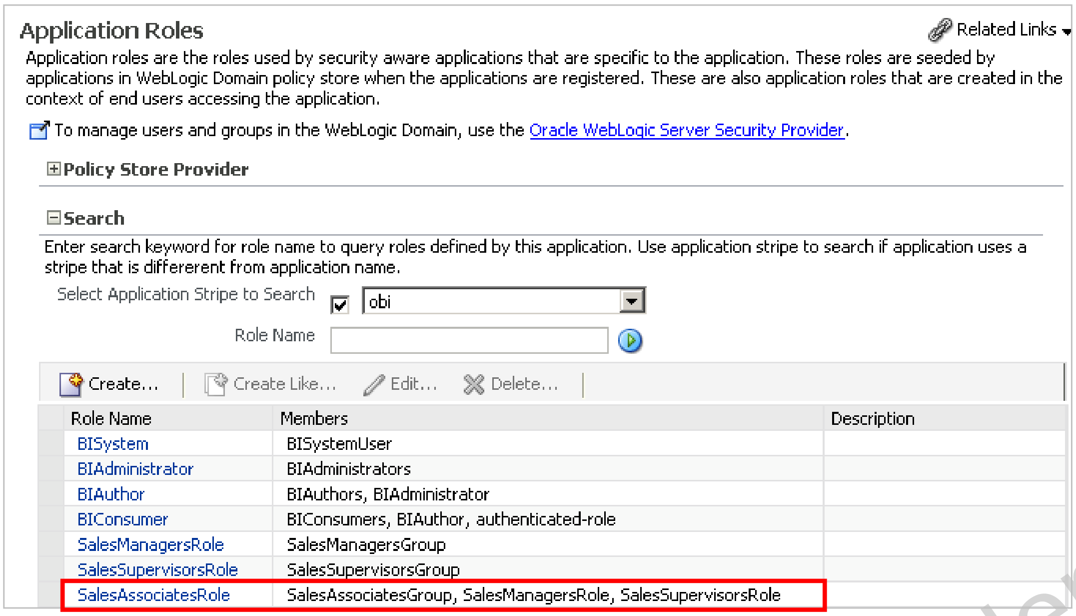
Map Application Roles

Once an application role is created, you can map the application role to users or groups defined in the LDAP server, or you can map the application role to other application roles.

In this example, the SalesAssociatesRole application role is mapped to the Sales Associates group, the Sales Managers application role, and the Sales Supervisors application role. This means that any user who is a member of the selected group or application role is mapped to this application role and receives any privileges or permissions assigned to the application role. It is possible to add individual users to a role, but a best practice is to add groups or application roles, not individual users, to application roles.

Application Role Hierarchies

Mapping application roles to other application roles creates application role hierarchies



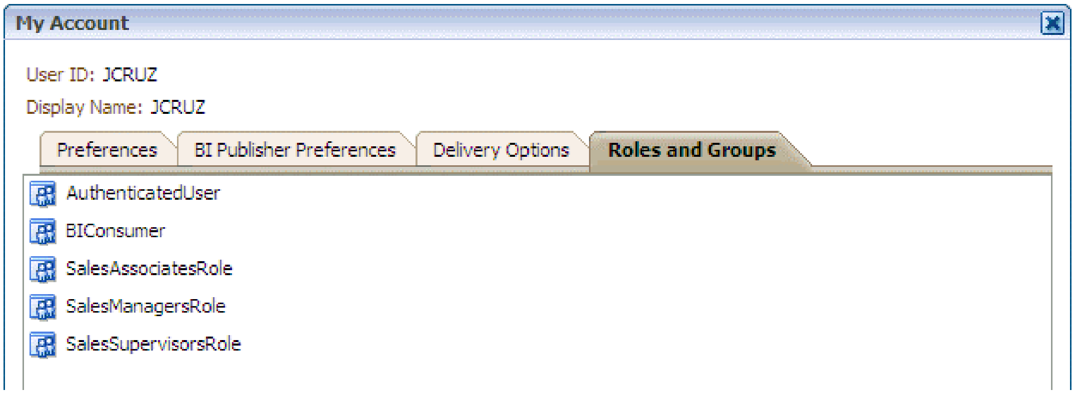
Application Role Hierarchies

Similar to the groups you created in the WebLogic identity store, you now have an application role hierarchy in the policy store. Mapping application roles to other application roles creates application role hierarchies. In this example, this means that any privileges or permissions assigned to the Sales Associates application role are inherited by the Sales Supervisors role and the Sales Managers role.

What is the difference between users and groups created in the identity store in the WebLogic LDAP Server and application roles created in the policy store in FMW Enterprise Manager? In the WebLogic LDAP server, you have users and groups. An application role is a logical role that can be used within the application to secure content in a way that is *independent* of any particular LDAP server and the users and groups within that LDAP server. Security rules are built using application roles. If the underlying LDAP environment changes, the security rules persist. In a different LDAP environment, where group or user names might be different, you could remap the application roles to different groups or users, and the BI security structure (built with application roles) would not be affected.

Verify Security Settings in Oracle BI

Restart Oracle BI Server to make policy store changes visible throughout Oracle BI.



ORACLE

Verify Security Settings in Oracle BI

To make policy store changes visible throughout Oracle BI, you must restart Oracle BI Server. Typically, you have a steady set of application roles, so the frequency of adding and deleting application roles in the policy store should be quite low. What is more common is mapping those roles to new groups and users in the identity store, which does not require a server restart. Oracle BI Server must be restarted only when application roles are added or deleted.

In this example, `JCRUZ` has logged in to Oracle BI and selected My Account. On the "Roles and Groups" tab, he sees all of the application roles to which he is assigned. Jose Cruz is a member of the Sales Managers group (which is a member of the Sales Managers application role) and a member of the Sales Supervisors group (which is a member of the Sales Supervisors application role). Because both of these roles are members of the Sales Associate role, he is also a member of that role. By default, all BI users are also members of the out-of-the-box application roles, AuthenticatedUser and BIConsumer.

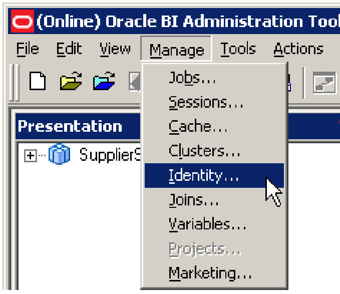
Verify Security Settings in Oracle BI (continued)

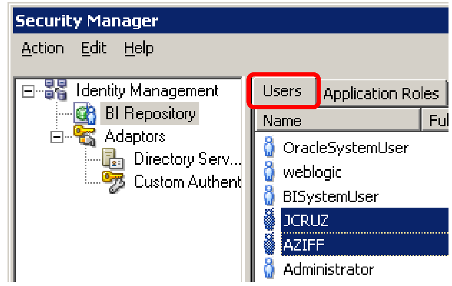
Application roles serve a variety of purposes in both development and production environments. In a development environment, developers can be granted one or more of the roles. One approach is to build roles that are eventually used in production, and then map developers to those roles for administering, building, and testing the development environment.

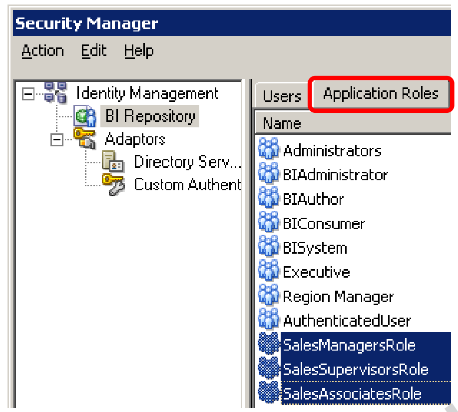
As you learn later in this lesson, you also use the logical application roles to secure access to repository objects and data. Therefore, application roles can be used to control access to both objects and functionality in the product.

The value of using application roles comes from the fact that you can move the system you have built between environments without having to rewire all of the security. For example, you would not have to change security settings in your presentation catalog or repository. You can just remap your application roles to the target environment.

Verify Security Settings in the Repository





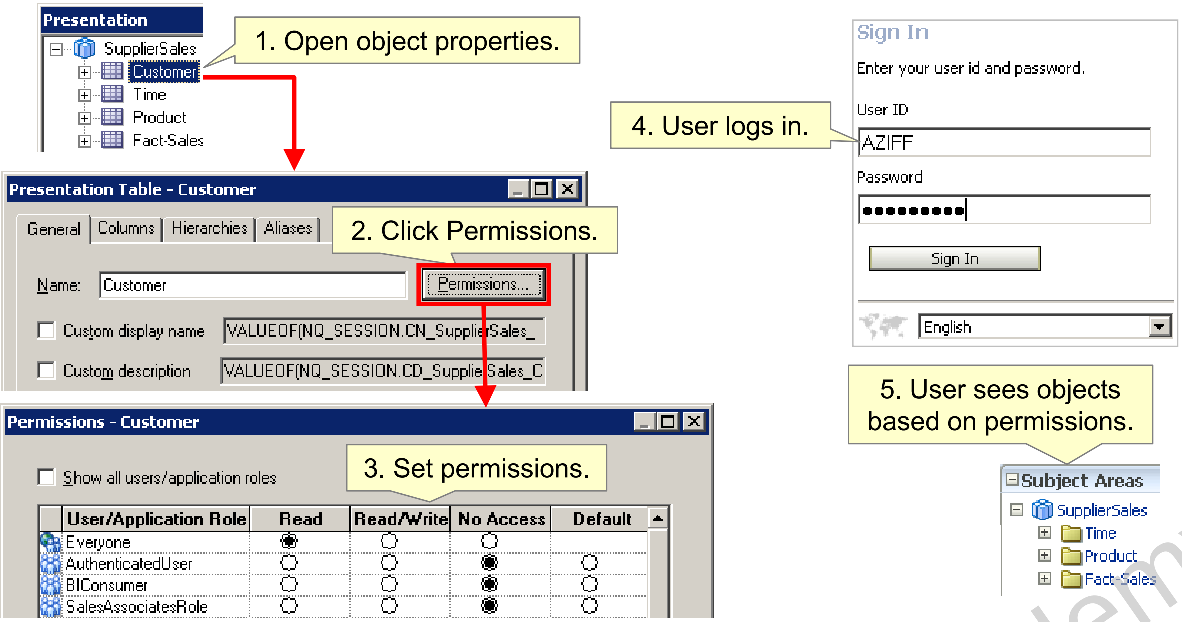


Verify Security Settings in the Repository

After you restart Oracle BI Server, changes in security settings should be visible in the Security Manager in the repository.

Set Up Object Permissions

Set up object permissions in your repository to control access to Presentation layer and BMM layer objects.



Set Up Object Permissions

You set object permissions by using the Administration Tool. There are two approaches to setting object permissions. You can set permissions for particular users or application roles in the Security Manager, or you can set permissions for individual objects in the Presentation layer. Setting up object permissions for individual users or application roles is useful when you want to define permissions for a large set of objects at one time. It is considered a best practice to set up object permissions for application roles rather than for individual users.

In this example, permissions are set for the `Customer` presentation table object. Access to this object is restricted for the AuthenticatedUser, BIConsumer, and SalesAssociatesRole application roles. The user `AZIFF` is a member of these application roles. Therefore, `AZIFF` does not have access to the `Customer` presentation table when he logs in to Oracle BI and selects the `SupplierSales` subject area.

Permission Inheritance

- Permissions granted explicitly to a user take precedence over privileges granted through application roles.
- Permissions granted explicitly to an application role take precedence over any privileges granted through other application roles.
- If security attributes conflict at the same level, a user or application role is granted the least-restrictive security attribute.

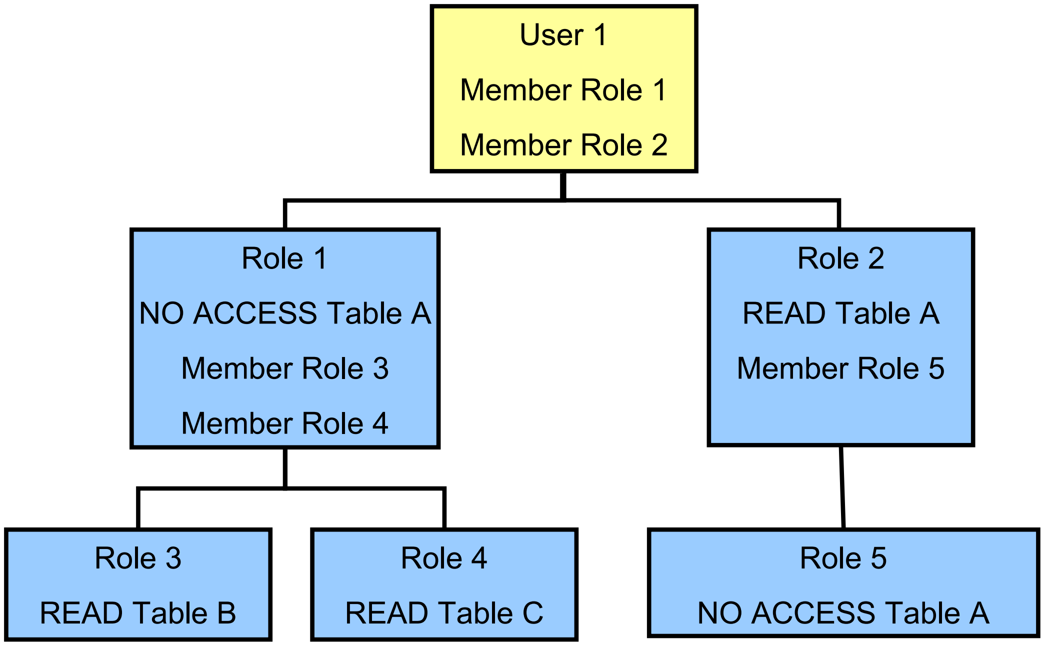
Permission Inheritance

Users can have explicitly granted permissions. They can also have permissions granted through membership in application roles, which in turn can have permissions granted through membership in other application roles, and so on. Permissions granted explicitly to a user take precedence over permissions granted through application roles, and permissions granted explicitly to the application role take precedence over any permissions granted through other application roles.

If there are multiple application roles acting on a user or application role at the *same level* with conflicting security attributes, the user or application role is granted the least-restrictive security attribute. Any explicit permissions acting on a user take precedence over any permissions on the same objects granted to that user through application roles.

Filter definitions, however, are always inherited. For example, if User1 is a member of Role1 and Role2, and Role1 includes a filter definition but Role2 does not, the user inherits the filter definition defined in Role1.

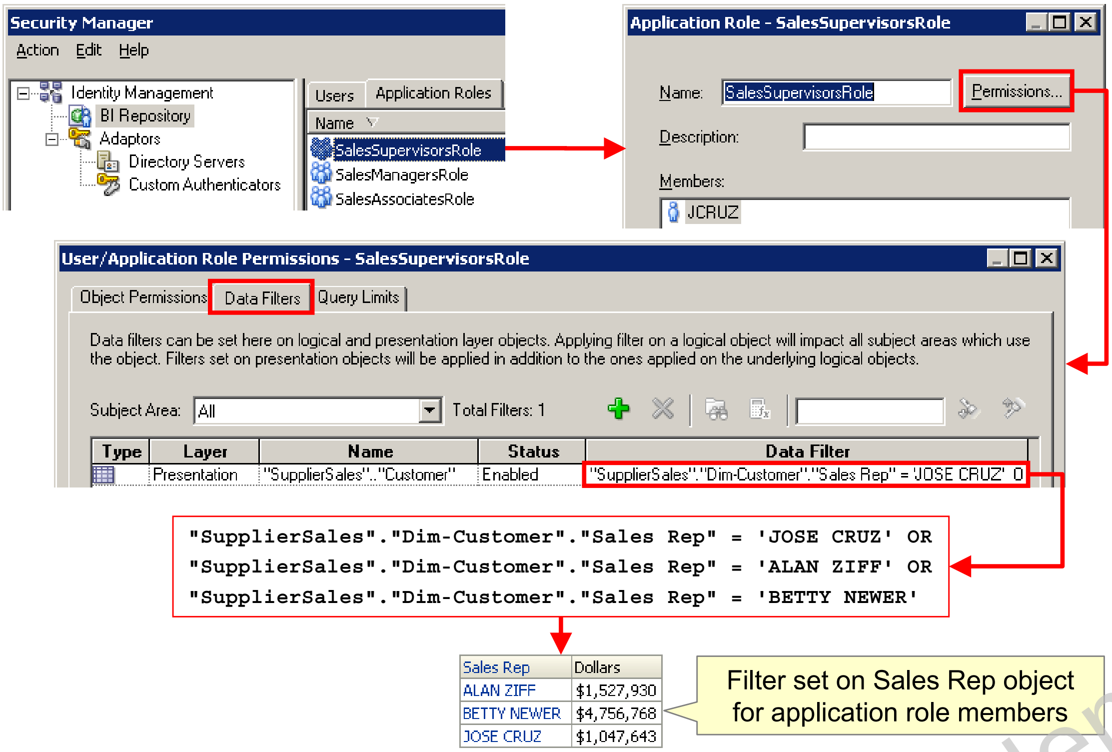
Permission Inheritance: Example



Permission Inheritance: Example

- User1 is a direct member of Role1 and Role2, and is an indirect member of Role3, Role4, and Role5.
- Because Role5 is at a lower level of precedence than Role2, its denial of access to `TableA` is overridden by the `READ` permission granted through Role2. The result is that Role2 provides `READ` permission on `TableA`.
- The resultant permissions from Role1 are `NO ACCESS` for `TableA`, `READ` for `TableB`, and `READ` for `TableC`.
- Because Role1 and Role2 have the same level of precedence and because the permission in each cancel the other out (Role1 denies access to `TableA`, Role2 allows access to `TableA`), the less-restrictive level is inherited by User1. That is, User1 has `READ` access to `TableA`.
- The total permissions granted to User1 are `READ` access for `TableA`, `TableB`, and `TableC`.

Set Row-Level Security (Data Filters)



Set Row-Level Security (Data Filters)

Data filters are a security feature that provides a way to enforce row-level security rules in the repository. Data filters are set up in the repository using the Administration Tool and are applied for a particular user or application role. You do not set up data filters if you have implemented row-level security in the database, because in this case, your row-level security policies are being enforced by the database rather than by Oracle BI Server.

Data filters can be set for objects in both the BMM layer and the Presentation layer. Applying a filter on a logical object affects all Presentation layer objects that use the object. If you set a filter on a Presentation layer object, it is applied in addition to any filters that might be set on the underlying logical objects.

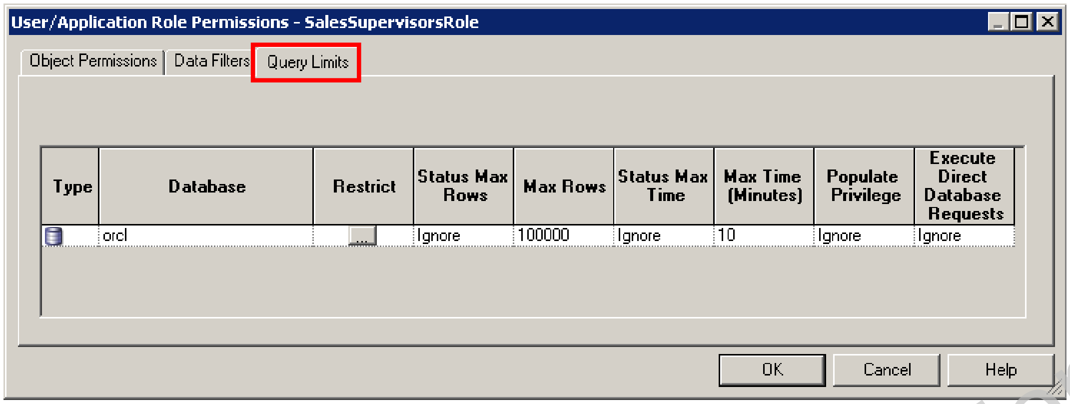
It is a best practice to set up data filters for particular application roles rather than for individual users.

In this example, you set a filter on the `Customer` presentation table for the SalesSupervisorsRole application role so that customer data is visible for only those records in which Jose Cruz or his direct reports are the sales representatives. Jose Cruz is a member of the SalesSupervisorsRole application role. After setting this filter, if Jose Cruz creates and runs an analysis that includes the Sales Rep column, he sees only his records and those of his direct reports (in this example, Alan Ziff and Betty Newer).

Set Query Limits

Use the Query Limits tab to:

- Control the number of rows accessed by a user or role
- Control the maximum query run time
- Enable or disable Populate Privilege
- Enable or disable Execute Direct Database Requests



Set Query Limits

Oracle BI Server enables you to exercise varying degrees of control over the repository information that users and roles can access. You can manage the query environment by setting query limits (*governors*) in the repository for users or application roles. You can prevent queries from consuming too many resources by limiting how long a query can run and how many rows a query can retrieve.

Note: It is a recommended practice to set query limits for application roles rather than for individual users.

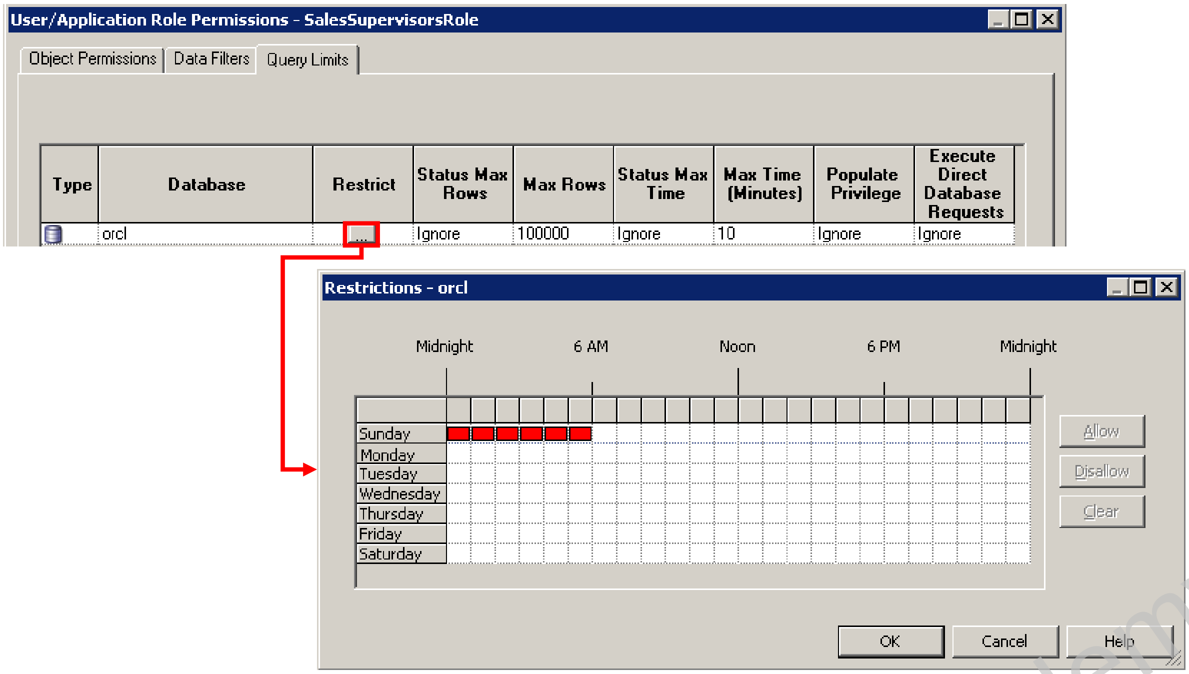
Use the Query Limits tab to control the following activities:

- Control runaway queries by limiting queries to a specific number of rows received by a user or role.
- Limit queries by maximum run time or to time periods for a user or role.
- Allow or disallow the populate privilege (this is primarily used for Marketing applications and is beyond the scope of this course).
- Allow or disallow execution of direct database requests for specific database objects.

To access the Query Limits tab, open the Security Manager, click the Application Roles tab, double-click an application role to open the Application Role dialog box, and click Permissions.

Set Timing Restrictions

Restrict access to a database during particular time periods.



Set Timing Restrictions

You can regulate when users can query databases to prevent users from querying when system resources are tied up with batch reporting, table updates, or other production tasks.

To restrict access to a database during particular time periods, click the ellipsis (**…**) button in the Restrict column to open the Restrictions dialog box. Then perform the following steps:

1. To select a time period, click the start time and drag it to the end time.
2. Access:
   - To explicitly allow access, click Allow.
   - To explicitly disallow access, click Disallow.

Summary

In this lesson, you should have learned how to:

- Identify and describe default security settings in Oracle BI
- Create users and groups
- Create application roles
- Set up permissions for repository objects
- Use query limits, timing restrictions, and filters to control access to repository information

Practice 19-1 Overview: Exploring Default Security Settings

This practice covers Oracle BI default security settings in the identity store, policy store, and credential store.

Practice 19-1 Overview: Exploring Default Security Settings

During installation, three Oracle BI security controls are preconfigured with initial (default) values to form the default security model. The security controls include:

- An embedded directory server functioning as an identity store designed to hold all user and group definitions that are required to control authentication
- A file-based policy store designed to hold the application role and permission grant mappings to users and groups that are required to control authorization
- A file-based credential store designed to hold all user and system credentials that are required to control authentication or authorization

Before you implement data access security in the Oracle BI repository, you explore these default security settings.

Practice 19-2 Overview: Creating Users and Groups

This practice covers using the WebLogic Server Administration Console to create users and groups.

Practice 19-2 Overview: Creating Users and Groups

Groups are logical ordered sets of users. Managing a group is more efficient than managing a large number of users individually. Oracle recommends that you first organize all Oracle BI users into groups that make sense for your organization's goals and map application roles to the groups to convey system privileges. The out-of-the-box identity store provided for managing users and groups is Oracle WebLogic Server's embedded directory server.

In this practice, you use the WebLogic Server Administration Console to create users and groups.

Practice 19-3 Overview: Creating Application Roles

This practice covers using Enterprise Manager Fusion Middleware Control to create application roles in the policy store.

Practice 19-3 Overview: Creating Application Roles

An application role conveys its permission grants to the users, groups, and application roles mapped to that role. Being mapped to an application role establishes membership in the role. Binding the permission grants to the application role streamlines the process of granting system privileges. After the application role and permission grant definitions are established, you control system rights by managing membership in each role.

Oracle recommends that you map groups to application roles and not to individual users. After they are mapped, all members of the group are granted the same rights. Controlling membership in a group reduces the complexity of tracking access rights for multiple individual users.

In this practice, you use Enterprise Manager Fusion Middleware Control to create application roles.

Practice 19-4 Overview: Setting Up Object Permissions

This practice covers setting up object permissions in the repository.

Practice 19-4 Overview: Setting Up Object Permissions

You can set up object permissions in your repository to control access to Presentation layer and BMM layer objects. You set object permissions by using the Administration Tool.

There are two approaches to setting object permissions. You can set permissions for particular users or application roles in the Security Manager, or you can set permissions for individual objects in the Presentation layer. In this practice you use both approaches.

Setting up object permissions for individual users or application roles is useful when you want to define permissions for a large set of objects at one time. It is a best practice to set up object permissions for particular application roles rather than for individual users.

Practice 19-5 Overview: Setting Row-Level Security (Data Filters)

This practice covers setting row-level security in the repository.

Practice 19-5 Overview: Setting Row-Level Security (Data Filters)

Data filters provide a way to enforce row-level security rules in the repository. Data filters are set up in the repository by using the Administration Tool and are applied for a particular user or application role.

Data filters can be set for objects in both the BMM layer and the Presentation layer. Applying a filter on a logical object affects all Presentation layer objects that use the object. If you set a filter on a Presentation layer object, it is applied in addition to any filters that might be set on the underlying logical objects. It is a best practice to set up data filters for particular application roles rather than for individual users.

In this practice, you set a filter on the `Customer` presentation table so that customer data is visible for only those records in which Jose Cruz or his direct reports are the sales representatives.

Practice 19-6 Overview: Setting Query Limits and Timing Restrictions

This practice covers managing the query environment by setting query limits and timing restrictions in the repository.

Practice 19-6 Overview: Setting Query Limits and Timing Restrictions

You can manage the query environment by setting query limits (governors) in the repository for users or application roles. You want to prevent queries from consuming too many resources by limiting how long a query can run and how many rows a query can retrieve. You also want to regulate when individual users can query databases to prevent users from querying when system resources are tied up with batch reporting, table updates, or other production tasks.

In this practice, you set the maximum rows of any query to five rows, the maximum time of any query to one minute, and restricted access to a database on Sunday from 12:00 AM to 7:00 AM.

20

Cache Management

Objective

After completing this lesson, you should be able to manage the Oracle BI Server query cache.

Business Challenge

- Decision support systems require a large amount of database processing.
- Frequent trips to back-end databases to satisfy query requests can result in:
  - Increased query response time
  - Poor performance

Business Challenge

Decision support queries sometimes require a large amount of database processing. Requests for the same information require frequent trips to back-end databases to retrieve the query results. Such trips can increase query response time and result in poor performance from the user's perspective.

Business Solution: Oracle BI Server Query Cache

Oracle BI Server can be configured to maintain a disk-based cache of query result sets (query cache):

- Saves the results of queries in cache files
- Enables Oracle BI Server to satisfy subsequent query requests without having to access back-end databases
- Improves query response time

Business Solution: Oracle BI Server Query Cache

Oracle BI Server can save the results of a query in cache files and then reuse them later when a similar query is requested. By using the cache, the database needs to be processed only once for the initial time a query is executed. For subsequent times that a similar query is executed, the results are satisfied by the cache and not the database.

Oracle BI administrators can configure Oracle BI Server to maintain a local, disk-based cache of query result sets (the *query cache*). The query cache enables Oracle BI Server to satisfy many subsequent query requests without having to access back-end databases. This reduction in communication costs can dramatically decrease query response time.

Advantages of Caching

- Eliminates unnecessary database processing because pre-computed results are stored in a local cache
- Improves query performance by fulfilling a query from the cache rather than searching through the database
- Conserves network resources by avoiding a connection to the database server

Advantages of Caching

The fastest way to process a query is to skip the bulk of the processing and use a pre-computed answer. With query caching, Oracle BI Server stores the pre-computed results of queries in a local cache. If another query can use those results, all database processing for that query is eliminated. This can result in dramatic improvements in the average query response time. Not running the query on the database also frees the database server to do other work.

In addition to improving performance, being able to answer a query from a local cache conserves network resources and processing time on the database server. Network resources are conserved because the intermediate results do not have to come over the network to Oracle BI Server. Administrators can take additional steps to improve caching performance. Such measures might include tuning and indexing databases, optimizing data source connectivity, leveraging aggregate tables, and constructing metadata efficiently.

Costs of Caching

- Disk space
  - The query cache requires dedicated disk space.
- Administrative tasks:
  - Set the cache persistence time appropriately.
  - Purge the cache when necessary.
- Keeping the cache up to date:
  - Evaluate what level of noncurrent information is acceptable.
  - Remove stale data.

Costs of Caching

Disk Space

The query cache requires dedicated disk space. The amount of space depends on the query volume, the size of the query result sets, and the amount of disk space you choose to allocate to the cache. For performance purposes, a disk should be used exclusively for caching, and it should be a high-performance, high-reliability disk system.

Administrative Tasks

There are a few administrative tasks associated with caching. You need to set the cache persistence time for each physical table appropriately, knowing how often data in that table is updated. When the frequency of the update varies, you need to keep track of when changes occur and purge the cache manually when necessary.

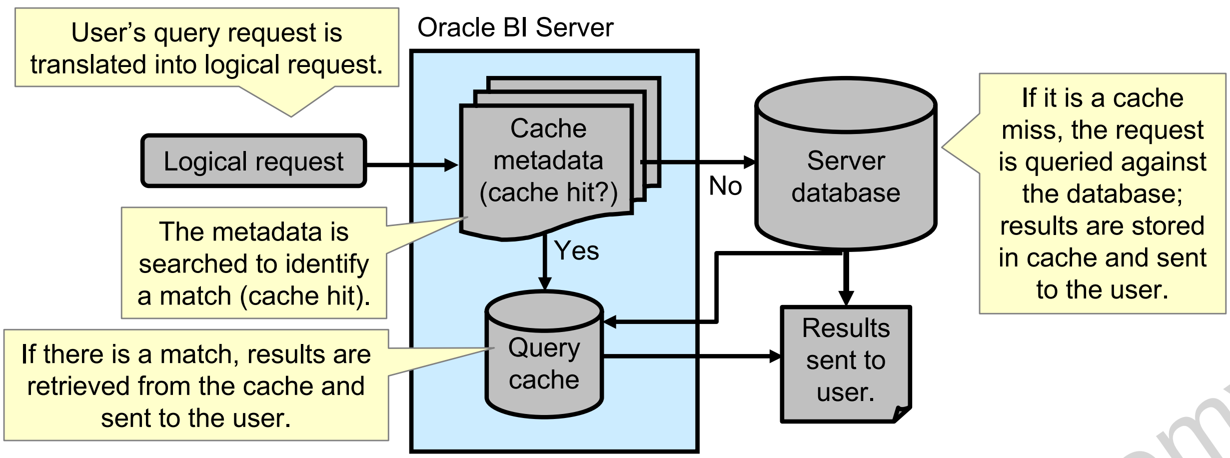
Keeping the Cache Up to Date

If the cache entries are not purged when the data in the underlying databases changes, queries can potentially return results that are out of date. You need to evaluate whether this is acceptable. It might be acceptable to allow the cache to contain some stale data. You need to decide what level of stale data is acceptable and then set up (and follow) rules to reflect those levels. For applications in which data is updated yearly or quarterly, it may be acceptable to keep stale data in the cache. For applications in which data is updated frequently, it may be necessary to purge the cache more often. It is also possible to purge the entire cache as part of the extraction, transformation, and loading (ETL) process for rebuilding the data mart, making sure that there is no stale data in the cache.

Query Cache: Architecture

The query cache consists of:

- Cache storage space
- Cache metadata
- Cache detection



Query Cache: Architecture

A query cache is a facility that stores the results from queries. If a query is fulfilled by the results stored in the cache, it is called a *cache hit*. A cache hit means that the server was able to use the cache to answer the query and did not go to the database at all.

A cache is used to eliminate redundant queries. For example, if 10,000 users always look at the same dashboard, getting the data once and storing it in the cache helps with scalability.

This graphic in the slide depicts the basic architecture of the query cache. The process of accessing the cache metadata occurs very quickly. If the metadata shows a cache hit, the bulk of the query processing is eliminated, and the results are immediately returned to the user. The process of adding the new results to the cache is independent of the results being returned to the user; the only effect on the running query is the resources consumed in the process of writing the cached results.

Monitoring and Managing the Cache

Requires an overall cache management strategy:

- Invalidate the cache entries when underlying data has changed.
- Monitor, identify, and remove undesirable cache entries.

Monitoring and Managing the Cache

Cache files always produce the same results, even after a database has been updated. Issues with retaining cache files may arise. For example, not purging outdated caches (known as *stale caches*) can potentially return inaccurate results over time and consume disk space.

Therefore, you need a cache management strategy to manage changes to the underlying databases and to monitor cache entries. The choice of a cache management strategy depends on the volatility of the data in the underlying databases and the predictability of the changes that cause this volatility. It also depends on the number and types of queries that comprise your cache, as well as the usage those queries receive. Cache management techniques are provided in the following slides.

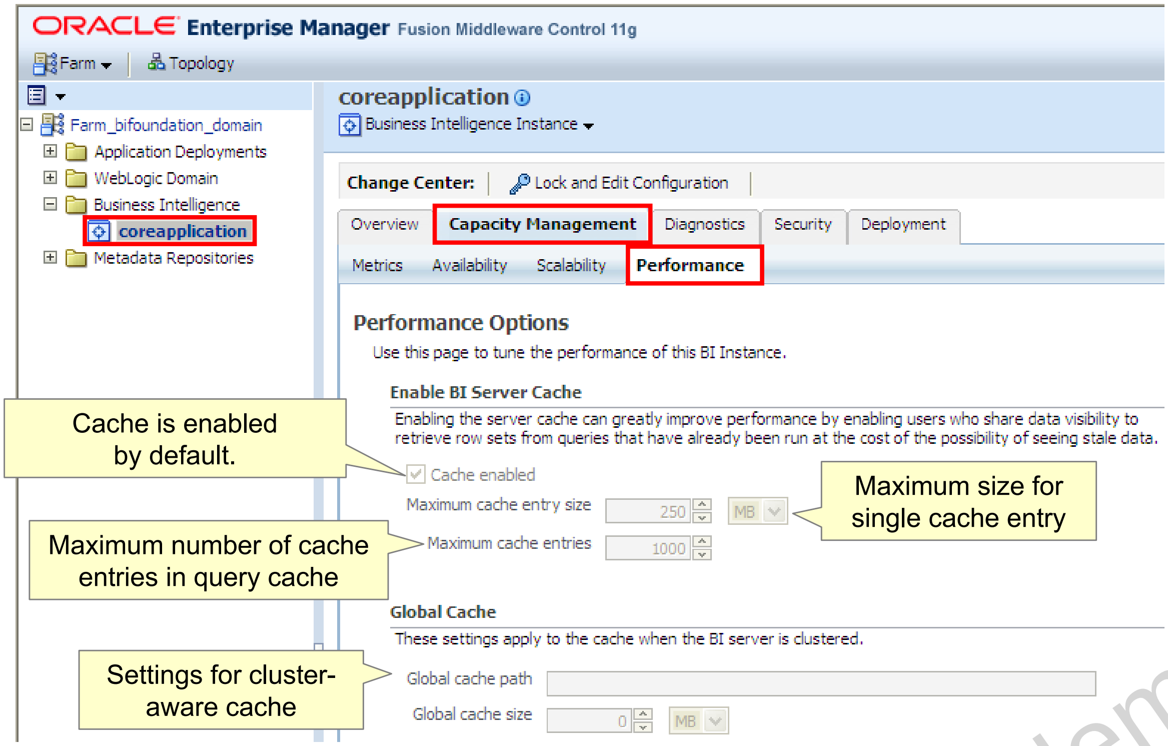
Cache Management Techniques

- Using Fusion Middleware Control to configure caching
- Using `NQSConfig.ini` to manually edit cache parameters
- Setting caching and cache persistence for tables
- Using the Cache Manager
- Inspecting SQL for cache entries
- Modifying the Cache Manager column display
- Inspecting the cache reports
- Purging cache entries manually using the Cache Manager
- Purging cache entries automatically
- Using event polling tables
- Seeding the cache

Cache Management Techniques

This slide lists some of the techniques you can use to manage the query cache. Each technique is discussed in detail in subsequent slides.

Using Fusion Middleware Control to Configure Caching



Using Fusion Middleware Control to Configure Caching

You can use Fusion Middleware Control Enterprise Manager to enable or disable query caching. The query cache is enabled by default. To use Enterprise Manager to enable or disable query caching, go to the Business Intelligence Overview page, display the Performance tab of the Capacity Management page, click "Lock and Edit Configuration," and select "Cache enabled." To disable query caching, deselect "Cache enabled." Apply and activate your changes.

You also can use Fusion Middleware Control to set the maximum size for a single cache entry, as well as the maximum number of cache entries in the query cache. When the cache storage directories exceed the number specified in the `Maximum cache entries` parameter, the entries that are least recently used (LRU) are discarded to make space for new entries.

In a clustered environment, Oracle BI Server can be configured to access a shared cache that is referred to as the *cluster-aware cache*. This cluster-aware cache, residing on a shared file-system storage device, stores seeding and purging events as well as the result sets associated with the seeding events. You specify the physical location for storing cache entries shared across the cluster, as well as the maximum size for each global cache entry, using Fusion Middleware Control.

The cluster-aware cache is not covered in greater detail in this course. For more information about caching in a clustered environment, see "About the Cluster-Aware Cache" in the *System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

Using NQSConfig.ini to Manually Edit Cache Parameters

Specifies directories for query cache storage

Controls maximum number of rows for any cache entry

Rolls up aggregate from a previously executed query

```
[CACHE]

ENABLE = YES;
# A comma separated list of <directory maxSize> pair(s).
# These are relative to the process instance directory.
#    e.g. DATA_STORAGE_PATHS = "nQSCache" 500 MB;
#    resolves to
#
$(ORACLE_INSTANCE)/bifoundation/OracleBIServerComponent/
<instance_name>/nQSCache
DATA_STORAGE_PATHS = "cache" 500 MB;
MAX_ROWS_PER_CACHE_ENTRY = 50000;  # 0 is unlimited size
MAX_CACHE_ENTRY_SIZE = 1 MB;
MAX_CACHE_ENTRIES = 500;
POPULATE_AGGREGATE_ROLLUP_HITS = NO;
USE_ADVANCED_HIT_DETECTION = NO;

MAX_SUBEXPR_SEARCH_DEPTH = 7;
DISABLE_SUBREQUEST_CACHING = NO;
# Cluster-aware cache.
# Note that since this is a network share, the directory
should not be
# relative.

GLOBAL_CACHE_STORAGE_PATH = "" 0;
MAX_GLOBAL_CACHE_ENTRIES = 1000;
CACHE_POLL_SECONDS = 300;
CLUSTER_AWARE_CACHE_LOGGING = NO;
```

ORACLE

Using NQSConfig.ini to Manually Edit Cache Parameters

Cache parameters modified in Fusion Middleware Control are stored in the [CACHE] section of the NQSConfig.ini configuration file. You can set additional query cache parameters in the NQSconfig.ini file, including the following:

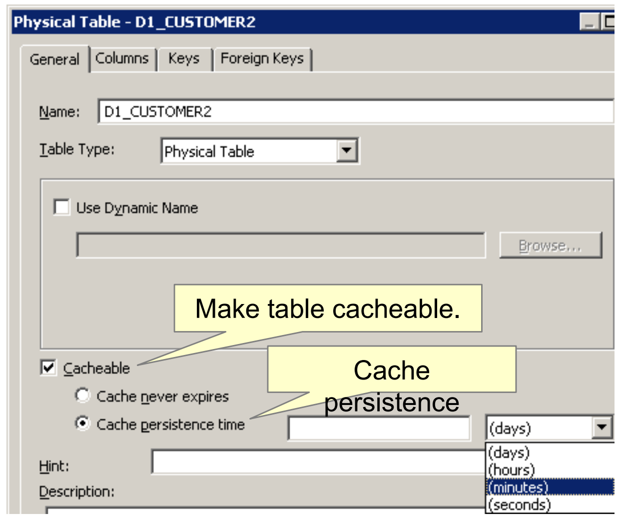
DATA_STORAGE_PATHS : Specifies one or more directories for query cache storage, as well as the maximum size for each storage directory. These directories are used to store the cached query results and are accessed when a cache hit occurs. The cache storage directories should reside on high-performance storage devices, ideally devoted solely to cache storage. When the cache storage directories begin to fill up, the entries that are least recently used (LRU) are discarded to make space for new entries.

MAX_ROWS_PER_CACHE_ENTRY: Controls the maximum number of rows for any cache entry. Limiting the number of rows is a useful way to avoid using up the cache space with runaway queries that return large numbers of rows. If the number of rows that a query returns is greater than the value specified in the MAX_ROWS_PER_CACHE_ENTRY parameter, the query is not cached.

POPULATE_AGGREGATE_ROLLUP_HITS: Typically, if a query gets a cache hit from a previously executed query, the new query is not added to the cache. When set to YES, this parameter overrides this default when the cache hit occurs by rolling up an aggregate from a previously executed query.

Setting Caching and Cache Persistence for Tables

- Enable caching for a table so that any query involving the table is added to the cache.
  - All tables are cacheable by default.
- Set cache persistence time to indicate how long entries for this table should be kept in the cache.



Setting Caching and Cache Persistence for Tables

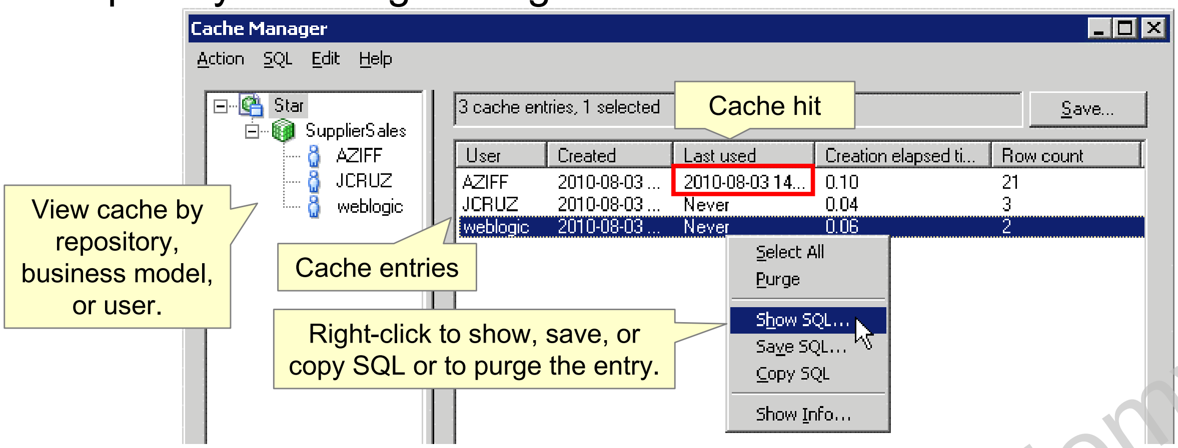
You can set a cacheable attribute for each physical table, enabling you to specify whether queries for that table are added to the cache to answer future queries. If you enable caching for a table, any query involving the table is added to the cache.

All tables are cacheable by default, but some tables may not be good candidates to include in the cache unless you use the cache persistence time settings. For example, suppose that you have a table that stores stock ticker data that is updated every minute. You can use the cache persistence time settings to purge the entries for that table every 59 seconds. You can also use the “Cache persistence time” field to specify how long the entries for this table should be kept in the query cache. This is useful for data sources that are updated frequently.

In other cases, however, it may not make sense to produce a cache at all when a table is queried against. In this case, an administrator can indicate that a table is non-cacheable by deselecting the Cacheable check box. With a table that is non-cacheable, any query against that table is not added to the cache. Defining tables as non-cacheable is generally done only when there are tables that change very frequently (such as a real-time subject area).

Using the Cache Manager

- View information about the entire query cache or individual cache entries.
- Show, save, or copy cache SQL.
- Manually purge the cache entries.
- Open by selecting Manage > Cache in online mode.



Using the Cache Manager

The Cache Manager can be viewed in online mode only. Select Manage > Cache to open the Cache Manager.

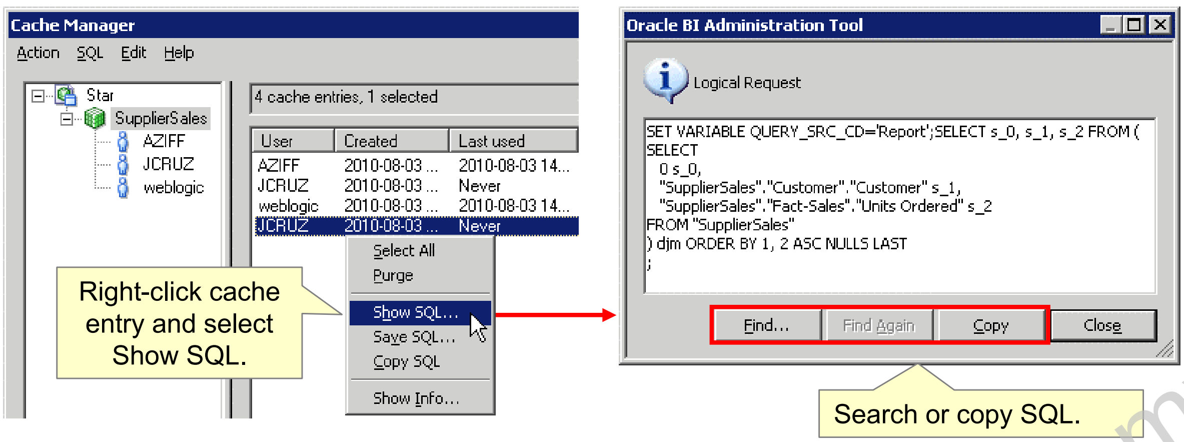
The Cache Manager provides Oracle BI Server administrators the ability to view information about the entire query cache, as well as information about individual entries in the query cache associated with the open repository. It also provides the ability to select specific cache entries and perform various operations on those entries (such as viewing and saving the cached SQL call) or purge them.

Click the Cache tab at the bottom of the left pane (not shown here) to view the cache entries for the current repository, business models, and users. The associated cache entries are reflected in the right pane, with the total number of entries shown in the View-Only field at the top. Cache hits are shown in the "Last used" column.

Inspecting SQL for Cache Entries

The Cache Manager provides the ability to view the SQL for cache entries in a separate window to:

- Evaluate SQL of queries that receive frequent hits
- Search and troubleshoot SQL

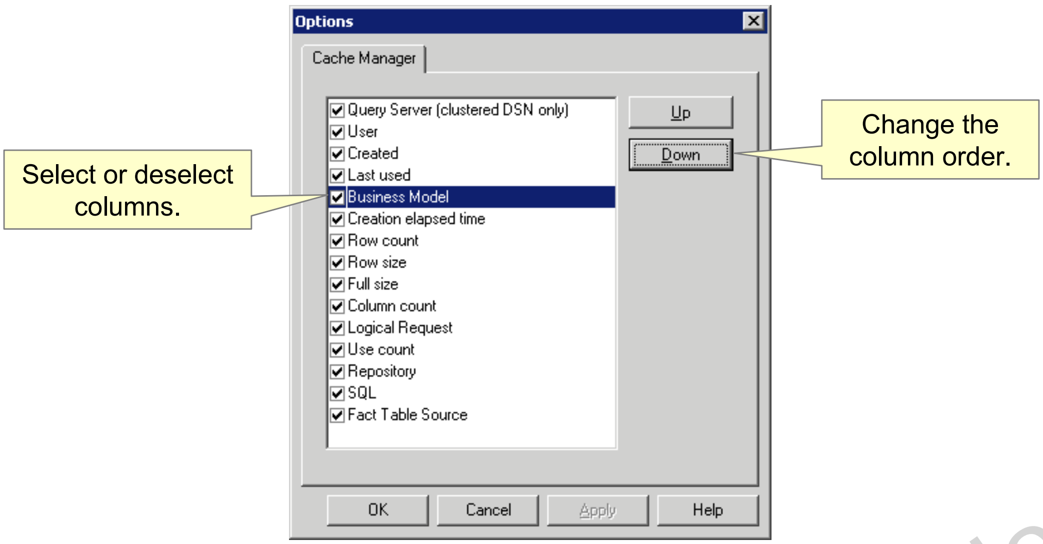


Inspecting SQL for Cache Entries

The Cache Manager provides the ability to view the SQL for a cache entry in a separate window. To see the logical SQL used by the query, right-click the cache entry and select Show SQL or select the cache entry and select SQL > Show from the menu. The Find button and Find Again button enable you to search and troubleshoot complex queries. The SQL for a request can assist in evaluating cache statistics. For example, you realize that a cache entry has fulfilled 90 other requests, and you may want to know the SQL behind this entry to seed the cache requests that are just as effective.

Modifying the Cache Manager Column Display

1. Select Edit > Options.
2. Deselect a column to remove it from the display.
3. Use the Up and Down buttons to change the column order.



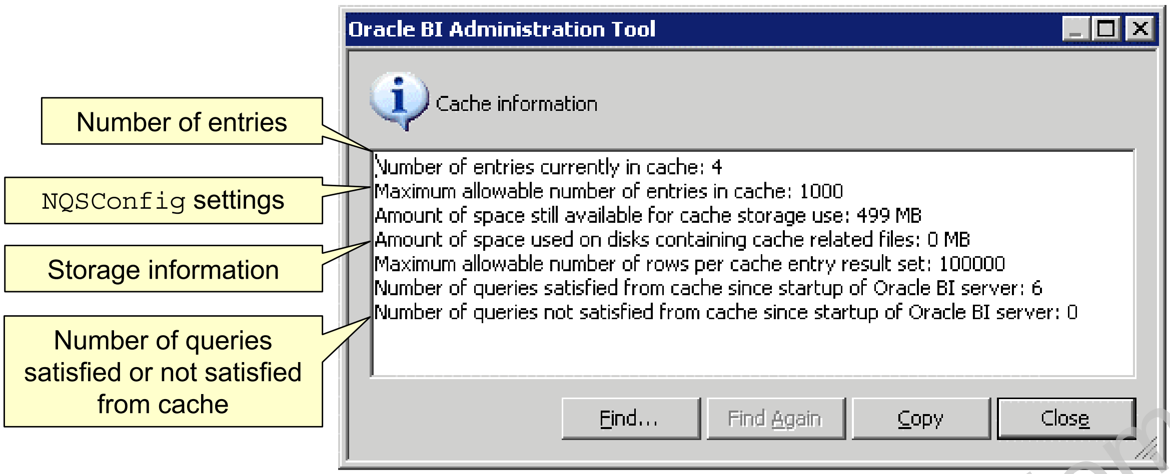
Modifying the Cache Manager Column Display

You can alter how the Cache Manager displays information. The Options window allows you to choose the columns that you want the Cache Manager to display by selecting or deselecting the check boxes for the columns. You can also use the Up and Down buttons to set the order in which columns are displayed.

Inspecting Cache Reports

Select Action > Show Info to display global cache information:

- Query information
- Storage information
- `NQSConfig.ini` cache settings



Inspecting Cache Reports

In the Cache Manager, select Action > Show, or right-click in the right pane and select Show to display global cache information. This includes information such as:

- Number of entries currently in the cache
- Number of queries satisfied by the cache
- Number of queries not satisfied

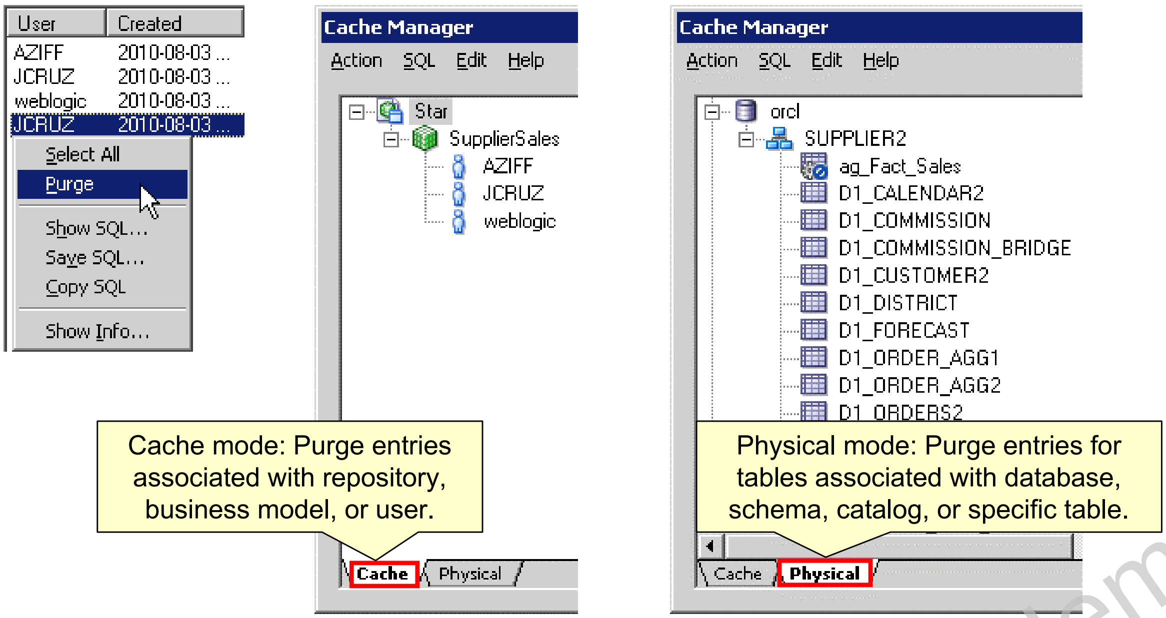
The report also includes cache settings, such as:

- Maximum allowable number of entries in the cache
- Storage space information
- Maximum allowable number of rows per cache entry result set

Administrators can use this information to monitor cache performance. For example, if a large number of queries are not being satisfied by the cache, this could affect overall performance.

Purging Cache Entries Manually Using the Cache Manager

Purging is the process of deleting entries from the query cache.



Purging Cache Entries Manually Using the Cache Manager

Purging cache is the process of deleting entries from the query cache. There are two methods for manually deleting cache entries:

- **Cache mode:** Purge one or more selected cache entries associated with the open repository, a specified business model, or a specified user within a business model.
- **Physical mode:** Purge all cache entries for tables associated with one or more selected databases, one or more selected catalogs, one or more selected schemas, or all cache entries associated with one or more selected tables.

To purge cache entries, select the entries and then right-click and select purge, or select Edit > Purge.

Purging cache entries manually gives the administrator the highest level of control over purging but is not necessarily the most efficient method. Automated methods for purging cache are discussed in the next slide.

Purging Cache Entries Automatically

Cache entries can also be purged automatically by:

- Setting the Cache persistence time field for a physical table
- Filling up the allotted cache storage space
- Setting up an event polling table

Purging Cache Entries Automatically

As you learned earlier in this lesson, cache is automatically purged if certain conditions are met. You can set cache persistence time to indicate how long entries for this table should be kept in cache. You can also use cache parameters in `NQSConfig.ini`, such as `MAX_CACHE_ENTRIES`, to limit the total number of cache entries. When cache storage exceeds the specified number, entries that are least recently used are discarded, which essentially purges the cache.

Another technique for automatically purging the cache is the use of event polling tables. This is discussed in more detail in the next slide.

Using Event Polling Tables

- Event polling tables store information about updates in the underlying databases.
- Oracle BI Server polls tables at set intervals and purges any stale cache entries that reference the updated tables.
- Using event polling tables can be the sole method of cache management or can be used with other cache-management techniques.
- Event tables require a fixed schema.

Caution: Because there is a polling interval in which the cache is not completely up to date, there is always the potential for stale data in the cache.

Using Event Polling Tables

An application (such as an application that loads data into a data mart) can be configured to add rows to an event polling table each time a database table is updated. Oracle BI Server polls this table at set intervals and purges any stale cache entries that reference the tables.

The event table is a physical table that resides on a database accessible to Oracle BI Server. Regardless of where it resides—in its own database or in a database with other tables—it requires a fixed schema. It is normally exposed only in the Physical layer of the Administration Tool, where it is identified in the Physical Table dialog box as being an Oracle BI Server event table.

The use of event tables is one of the most accurate ways of invalidating stale cache entries, and it is probably the most reliable method. It does, however, require the event table to be populated each time a database table is updated. Also, because there is a polling interval in which the cache is not completely up to date, there is always the potential for stale data in the cache.

Note: Setting up polling tables is beyond the scope of this course.

Seeding the Cache

Seeding is the process of pre-populating the cache with queries that are known to generate cache hits.

- Helps improve query performance
  - Use queries that heavily consume database processing and are likely to be reused.
- Is performed by running prebuilt queries during off hours or immediately after purging
  - Manually by using the Oracle BI user interface
  - Automatically by using the Oracle BI scheduler to schedule queries to run at a specified time

Seeding the Cache

One of the main advantages of seeding the cache is the improvement of query performance. A good strategy, therefore, is to seed the cache during off hours by running queries and caching their results. A good seeding strategy requires knowing when cache hits occur so that you can seed the cache with the appropriate queries. The best queries for seeding the cache are queries that heavily consume database processing resources or that are likely to be reused.

For example, seed the cache with queries that have many joins or a great deal of sorting, or with queries that are used frequently throughout the business day. Be careful not to seed the cache with simple queries that return many rows and require very little database processing.

Cache Hit Conditions

A cache hit occurs only when certain conditions are met, such as the following:

- Query `WHERE` clause constraints need to be equivalent to the cached results or to a subset of the results.
- All the columns in the `SELECT` list of a new query must exist in the cached query or they must be able to be calculated from the columns in the query.
- Join conditions must be equivalent.
- Queries that request an aggregated level of information can use cached results at a lower level of aggregation.

Cache Hit Conditions

When caching is enabled, each query is evaluated to see if it qualifies for a cache hit. A *cache hit* means that Oracle BI Server was able to use the cache to answer the query and did not go to the database. The slide contains only a partial list of cache hit conditions. For a full list, see the *System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

Summary

In this lesson, you should have learned how to manage the Oracle BI Server query cache.

Practice 20-1 Overview: Enabling Query Caching

This practice covers the following topics:

- Enabling query caching
- Using the Cache Manager
- Using the query log to verify cache hits and non-hits

Practice 20-1 Overview: Enabling Query Caching

You use Fusion Middleware Control to enable query caching and to create and run analyses. You then use the Cache Manager and the query log to inspect cache entries and analyze cache hits and non-hits.

Practice 20-2 Overview: Modifying Cache Parameters

This practice covers the following topics:

- Modifying cache parameters
- Making tables non-cacheable
- Modifying Cache Manager display options
- Purging cache entries manually

Practice 20-2 Overview: Modifying Cache Parameters

You use the Cache Manager to inspect the cache parameters. You then modify the number of rows per cache as well as the number of cache entries allowed. In addition to modifying cache parameters, you make certain tables non-cacheable.

Practice 20-3 Overview: Seeding the Cache

This practice covers the following topics:

- Creating a query to seed the cache
- Creating, scheduling, and delivering an agent to seed the cache

Practice 20-3 Overview: Seeding the Cache

You have identified requests that are used frequently by sales representatives. To improve performance, you want to seed the cache with this data.

In this practice, you create and save a query to populate the cache and then you create an agent to seed the cache. During this process, you use a programmatic ODBC call to purge the cache.

21

Managing Usage Tracking

Objectives

After completing this lesson, you should be able to:

- Identify the need for usage tracking
- Set up and administer Oracle BI usage tracking

Business Challenges

- When it is first deployed, Oracle BI may not be optimized for the querying that actually occurs:
  - End-user queries may not match what is expected, so the cache is not seeded with appropriate queries.
  - Additional aggregate tables may need to be created to speed up query processing.
- Your company may need to track database usage on a user or departmental level:
  - Users may be charged for database use.
  - Regulatory requirements may require usage tracking.

Business Solution: Oracle BI Usage Tracking

- Tracks and stores Oracle BI Server usage at the detailed query level
- Supports the accumulation of usage tracking statistics that can be used in a variety of ways, such as:
  - Database performance optimization
  - Aggregation strategies
  - Billing users or departments based on the resources they consume
- Provides ability to analyze usage results with end-user reporting tools

Oracle BI Usage Tracking Methods

There are two methods for enabling usage tracking:

- Direct insertion (recommended approach)
  - Oracle BI Server inserts statistics for every query directly into a relational database table.
- Log file
  - Oracle BI Server inserts statistics for every query into a log file.

Oracle BI Usage Tracking Methods

When you enable usage tracking, statistics for every query are inserted into a database table or are written to a usage tracking log file. If you use direct insertion, Oracle BI Server directly inserts the usage tracking data into a relational database table. It is recommended that you use direct insertion to write statistics to a database table.

Only the direct insertion method is discussed in this lesson. For more information about setting up a log file to collect usage tracking information, see the *System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

ABC Example

Set up Oracle BI usage tracking to track and store usage statistics at the detailed query level.

Username | Date | Time | Subject area | Logical SQL | Row count

User Name	Date	Time	Subject Area Name	Logical SQL	Max Row Count
AZIFF	8/3/2010 12:00:00 AM	17:03	SupplierSales	SELECT s_0, s_1, s_2, s_3 FROM (SELECT 0 s_0, "SupplierSales"."Product"."Specific" s_1, "SupplierSales"."Fact-Sales"."Dollars" s_2, "SupplierSales"."Fact-Sales"."Units Ordered" s_3 FROM "SupplierSales") djm ORDER BY 1, 2 ASC NULLS LAST	181
JCRUZ	8/3/2010 12:00:00 AM	17:04	SupplierSales	SELECT s_0, s_1, s_2 FROM (SELECT 0 s_0, "SupplierSales"."Time"."Date" s_1, "SupplierSales"."Fact-Sales"."Dollars" s_2 FROM "SupplierSales") djm ORDER BY 1, 2 ASC NULLS LAST	400

ORACLE

Steps to Enable Usage Tracking

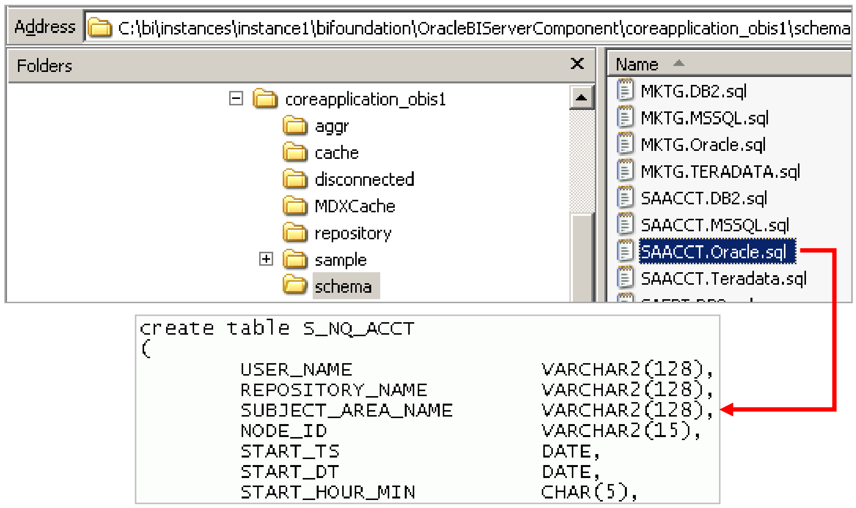
1. Create the usage tracking table.
2. Import the usage tracking table.
3. Build a usage tracking business model.
4. Enable usage tracking.
5. Enable direct insertion.
6. Set the physical table parameter.
7. Set the connection pool parameter.
8. Set additional parameters.
9. Test the results.

Steps to Enable Usage Tracking

The steps in the following slides enable usage tracking using the direct insertion method.

1. Create the Usage Tracking Table

- Use the provided `SAACCT.<db>.sql` script to create the `S_NQ_ACCT` usage tracking table.
- This table stores the usage tracking data when queries are run against Oracle BI Server.



1. Create the Usage Tracking Table

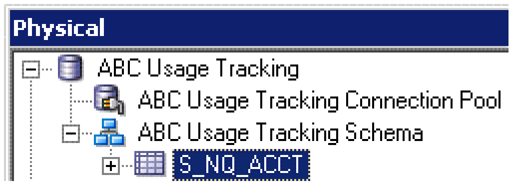
The following folder contains `CREATE TABLE` scripts for Oracle, DB2, SQL Server, and Teradata:

`<ORACLE_INSTANCE>\bifoundation\OracleBIServerComponent\coreapplication_obis1\schema`

The sample scripts set the usage tracking table name to `S_NQ_ACCT`. For sites that have Oracle BI Applications, this is the name used in the Oracle BI repository. Sites that build their own repositories may change the name of the usage tracking table. The table name must match the name used in the corresponding repository.

2. Import the Usage Tracking Table

Import the usage tracking table into the Physical layer.

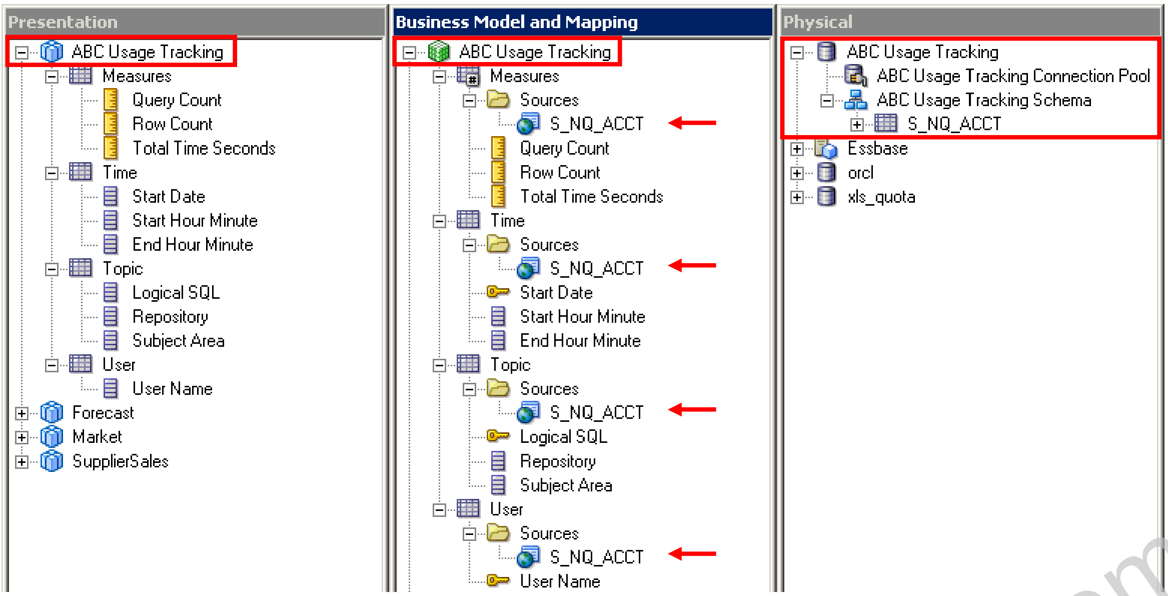


2. Import the Usage Tracking Table

Use known techniques to import the `S_NQ_ACCT` usage tracking table into the Physical layer of the repository. As shown in the screenshot, you can create a new database object, connection pool, and schema for usage tracking.

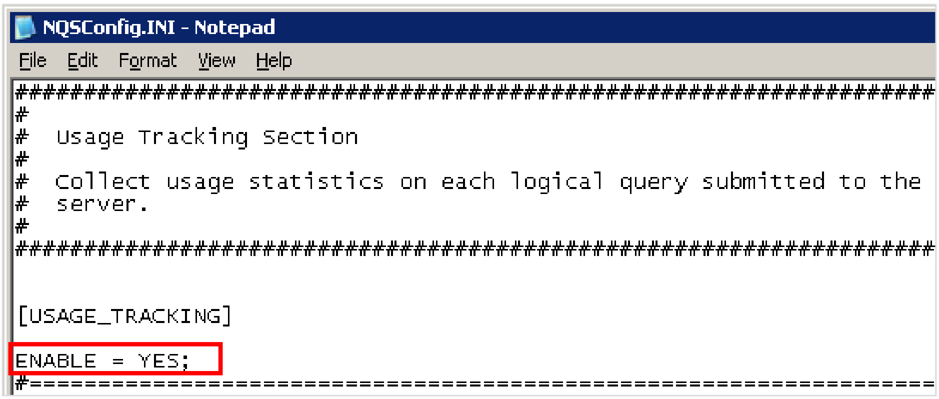
3. Build a Usage Tracking Business Model

Build a usage tracking business model using the columns in the `S_NQ_ACCT` table:



4. Enable Usage Tracking

Modify the `NQSConfig.ini` file to enable usage tracking.



NQSConfig.INI - Notepad

File Edit Format View Help

```
###############################################################################
#
#  Usage Tracking Section
#
#  Collect usage statistics on each logical query submitted to the
#  server.
#
###############################################################################


[USAGE_TRACKING]

ENABLE = YES;
#==============================================================================
```


4. Enable Usage Tracking

The `ENABLE` parameter in the `USAGE_TRACKING` section of the `NQSConfig.ini` file enables or disables collection of usage tracking statistics. Valid values are `YES` and `NO`. The default value is `NO`. When it is set to `NO`, statistics are not accumulated. When it is set to `YES`, statistics are accumulated for each logical query.

5. Enable Direct Insertion

Set the `DIRECT_INSERT` parameter to `YES` to specify that statistics are inserted directly into a database table.

NQSConfig.INI - Notepad

File Edit Format View Help

```
[USAGE_TRACKING]

ENABLE = YES;
#==============================================================================
# Parameters used for writing data to a flat file (i.e. DIRECT_INSERT = NO).
#
# Note that the directory should be relative to the instance directory.
# In general, we prefer directo insert to flat files.  If you are working in
# a cluster, it is strongly recommended you use direct insert.  If there is
# only one Oracle BI Server instance, then you may use flat file data.
# The directory is then assumed relative to the process instance.  For
# example, "UTData" is resolved to
# "$(ORACLE_INSTANCE)/bifoundation/OracleBIServerComponent/<instance_name>/UTData

STORAGE_DIRECTORY = "<directory path>";
CHECKPOINT_INTERVAL_MINUTES = 5;
FILE_ROLLOVER_INTERVAL_MINUTES = 30;
CODE_PAGE = "ANSI";  # ANSI, UTF8, 1252, etc.
#
#==============================================================================

DIRECT_INSERT = YES;
#==============================================================================
```

ORACLE

5. Enable Direct Insertion

In the `NQSConfig.ini` file, the `DIRECT_INSERT` parameter specifies whether statistics are inserted directly into a database table or written to a local file. When `DIRECT_INSERT` is set to `NO`, data is written to a flat file. When `DIRECT_INSERT` is set to `YES`, data is inserted into a table. Note that this parameter is operative only if `ENABLE = YES`. Direct insertion into a database table is recommended; therefore, the default value is `YES`.

When `DIRECT_INSERT` is set to `YES`, the following other parameters in `NQSConfig.ini` become valid: `PHYSICAL_TABLE_NAME`, `CONNECTION_POOL`, `BUFFER_SIZE`, `BUFFER_TIME_LIMIT_SECONDS`, `NUM_INSERT_THREADS`, and `MAX_INSERTS_PER_TRANSACTION`. These parameters are discussed in detail in the following slides.

When `DIRECT_INSERT` is set to `NO`, the following other parameters in `NQSConfig.ini` become valid: `STORAGE_DIRECTORY`, `CHECKPOINT_INTERVAL_MINUTES`, `FILE_ROLLOVER_INTERVAL_MINUTES`, and `CODE_PAGE`. These parameters are relevant only when the log file method of usage tracking is enabled; they are not discussed in this course. For more information about them, see the *System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

6. Set the Physical Table Parameter

- Set the PHYSICAL_TABLE_NAME parameter to specify the table into which to insert records corresponding to the query statistics.
- The table name is the fully qualified name as it appears in the Physical layer of the Administration Tool.

```
#============================================================================
DIRECT_INSERT = YES;

#============================================================================
#  Parameters used for inserting data into a table (i.e. DIRECT_INSERT = YES).
#
PHYSICAL_TABLE_NAME = "ABC Usage Tracking"."ABC Usage Tracking Schema"."S_NQ_ACCT";
CONNECTION_POOL = "ABC Usage Tracking"."ABC Usage Tracking Connection Pool";
BUFFER_SIZE = 250 MB;
BUFFER_TIME_LIMIT_SECONDS = 5;
NUM_INSERT_THREADS = 5;
MAX_INSERTS_PER_TRANSACTION = 1;
#
```


6. Set the Physical Table Parameter

In the NQSConfig.ini file, scroll to the “Parameters used for inserting data into a table” section and locate the PHYSICAL_TABLE_NAME parameter. To insert query statistics information into a table, you must provide the fully qualified physical table name of the table. The fully qualified physical table name consists of up to four components (database name, catalog name, schema name, and table name). Each component is surrounded by double quotes (") and separated by a period (.). This fully qualified physical table name must match a table name in the physical layer of the loaded repository.

Two format options are provided for the PHYSICAL_TABLE_NAME parameter. One has three components and one has four components. In the example in the slide, the fully qualified path to the usage tracking physical table consists of three components: a database component (ABC Usage Tracking), a schema component (ABC Usage Tracking Schema), and a physical table (S_NQ_ACCT).

7. Set the Connection Pool Parameter

- Set the `CONNECTION_POOL` parameter to specify the connection pool to use for inserting records into the usage tracking table.
- The connection pool is the fully qualified name as it appears in the Physical layer of the Administration Tool.

```
#==============================================================================

DIRECT_INSERT = YES;

#==============================================================================
#  Parameters used for inserting data into a table (i.e. DIRECT_INSERT = YES).
#
PHYSICAL_TABLE_NAME = "ABC Usage Tracking"."ABC Usage Tracking Schema"."S_NQ_ACCT";
CONNECTION_POOL = "ABC Usage Tracking"."ABC Usage Tracking Connection Pool";
BUFFER_SIZE = 250 MB;
BUFFER_TIME_LIMIT_SECONDS = 5;
NUM_INSERT_THREADS = 5;
MAX_INSERTS_PER_TRANSACTION = 1;
#
```


7. Set the Connection Pool Parameter

In the `NQSConfig.ini` file, set the `CONNECTION_POOL` parameter to specify the connection pool to use for inserting records into the usage tracking table. The fully-specified connection pool name has two parts: database object name and connection pool name. Each part is surrounded by double quotes (`"`) and separated by a period (`.`). In the example in the slide the parameter is the following:

```
"ABC Usage Tracking"."ABC Usage Tracking Connection Pool";
```

The fully qualified connection pool name should match a connection pool name in the physical layer of the loaded repository. For Usage Tracking inserts to succeed, the connection pool must be configured with a user ID that has write access to the back-end database.

8. Set Additional Parameters

- `BUFFER_SIZE`
  – Amount of memory that is used to store insert statements temporarily
- `BUFFER_TIME_LIMIT_SECONDS`
  – Maximum amount of time that an insert statement remains in the buffer before it is issued to the usage tracking table
- `NUM_INSERT_THREADS`
  – Number of threads that remove insert statements from the buffer and issue them to the usage tracking table
- `MAX_INSERTS_PER_TRANSACTION`
  – Number of records to group together as a single transaction when inserting into the usage tracking table

8. Set Additional Parameters

- `BUFFER_SIZE` specifies the amount of memory that is used to temporarily store insert statements. The buffer allows the insert statements to be issued to the usage tracking table independently of the query that produced the statistics to be inserted. When the buffer is full, subsequent query statistics are discarded until insert threads service the buffer entries.
  Example: `BUFFER_SIZE = 10 MB ;`
- `BUFFER_TIME_LIMIT_SECONDS` specifies the maximum amount of time that an insert statement remains in the buffer before it is issued to the usage tracking table. This time limit ensures that Oracle BI Server issues the insert statements in a timely manner even during periods of extended quiescence.
  Example: `BUFFER_TIME_LIMIT_SECONDS = 5 ;`
- `NUM_INSERT_THREADS` specifies the number of threads that remove insert statements from the buffer and issues them to the usage tracking table. The number of threads should not exceed the total number of threads assigned to the connection pool.
  Example: `NUM_INSERT_THREADS = 5 ;`
- `MAX_INSERTS_PER_TRANSACTION` specifies the number of records to group together as a single transaction when inserting into the usage tracking table. Increasing the number may slightly increase performance, but it also increases the possibility of inserts being rejected due to deadlocks in the database.
  Example: `MAX_INSERTS_PER_TRANSACTION = 1 ;`

9. Test the Results

Use the Usage Tracking subject area to build and run queries:

Users	Query Time		Topic		Measures
User Name	Date	Time	Subject Area Name	Logical SQL	Max Row Count

User Name	Date	Time	Subject Area Name	Logical SQL	Max Row Count
AZIFF	8/3/2010 12:00:00 AM	17:03	SupplierSales	SELECT s_0, s_1, s_2, s_3 FROM (SELECT 0 s_0, "SupplierSales"."Product"."Specific" s_1, "SupplierSales"."Fact-Sales"."Dollars" s_2, "SupplierSales"."Fact-Sales"."Units Ordered" s_3 FROM "SupplierSales") djm ORDER BY 1, 2 ASC NULLS LAST	181
JCRUZ	8/3/2010 12:00:00 AM	17:04	SupplierSales	SELECT s_0, s_1, s_2 FROM (SELECT 0 s_0, "SupplierSales"."Time"."Date" s_1, "SupplierSales"."Fact-Sales"."Dollars" s_2 FROM "SupplierSales") djm ORDER BY 1, 2 ASC NULLS LAST	400

Analyzing Usage Tracking Data

- Watch for long-running queries (typically ad hoc):
  - End users may need training.
  - You might need to assign query blocking or restrictions on how long queries can run or how many records are returned.
  - The database may require additional indexes or tuning.
- Perform usage audits for:
  - Regulatory compliance
  - Security
- Determine whether a query should be used to seed the cache or be removed from the cache-seeding queries.
- Identify aggregation strategies.
- Bill users or departments based on the resources that they consume.

Analyzing Usage Tracking Data

Why do you want to track usage?

It is helpful from an administrative and maintenance perspective to be able to identify:

- Users who may need training
- Better ways to bullet-proof the use of the application
- Candidate columns for indexing

Some organizations have strict security guidelines that require the ability to know who queried the data warehouse and when it was queried. In some cases, this information can be used to monitor and enforce regulatory compliance.

Queries that are being run and rerun may be candidates for cache seeding.

Summary

In this lesson, you should have learned how to:

- Identify the need for usage tracking
- Set up and administer Oracle BI usage tracking

Practice 21-1 Overview: Setting Up Usage Tracking

This practice covers setting up and administering usage tracking.

Practice 21-1 Overview: Setting Up Usage Tracking

Oracle BI Server supports the accumulation of usage tracking statistics that can be used in a variety of ways such as database optimization, aggregation strategies, or billing users or departments based on the resources they consume. Oracle BI Server tracks usage at the detailed query level.

In this practice, ABC wants to monitor the queries generated by users to help identify performance improvement areas. You use the recommended usage tracking approach, which is to track statistics by loading them directly into a database table rather than a log file. You use a provided script to create the necessary database table, modify the `NQSConfig.ini` file to support usage tracking, and test the results.

22

Setting Up and Using the Multiuser Development Environment

Objectives

After completing this lesson, you should be able to:

- Set up an Oracle BI multiuser development environment
- Describe the multiuser development environment functionality
- Develop a repository with multiple developers

Business Challenge

By default, the Oracle BI repository development environment is not set up for multiple users.

- Multiple developers working in online mode lock each other out as they check out objects.
- This causes inefficiency and potential conflicts while other developers wait for access to the repository.

Business Challenge

By default, the Oracle BI repository development environment is not set up for multiple users. Online editing makes it possible for multiple developers to work simultaneously. However, if they are working on the same business model, they lock each other out as they check out objects for editing.

This is an inefficient approach to repository development and can result in conflicts as developers may potentially overwrite each other's work. A more efficient development environment would permit developers to modify a repository simultaneously and then check in changes.

Business Solution: Oracle BI Multiuser Development Environment (MUDE)

- The Oracle BI MUDE permits multiple users to work with the repository simultaneously:
  - Users edit local copies of the repository.
  - Changes are saved locally and then merged to the master repository.
- The MUDE breaks the repository into manageable pieces known as *projects*.
  - Multiple users can work on the same or different projects.
  - Single users can improve efficiency by working on smaller subsets of the repository.

Business Solution: Oracle BI Multiuser Development Environment (MUDE)

Oracle BI enables multiple developers to work on repository objects from the same repository during group development of Oracle BI applications.

For example, after completing an implementation, the administrator might want to deploy Oracle BI to other functional areas of the company. In this example, multiple developers need to work concurrently on subsets of the metadata and merge these subsets back into a master repository without conflicts in the developers' work.

In another example, a single developer might manage all development. For simplicity and performance, this developer might want to use an Oracle BI MUDE to maintain metadata code in smaller chunks instead of in a large repository.

In both situations, this is accomplished by creating projects in the repository file in the Administration Tool and then copying this repository file to a shared network directory. Developers can check out projects, make changes, and then merge the changes into the master repository.

Oracle BI Repository Development Process

Adheres to the classic Software Configuration Management (SCM) process:

- It is conceptually and functionally analogous to processes found in pure-play source control systems.
- Developers can check out, work on, and merge from the master code repository.
- Oracle BI enables and manages checkout, merging, conflict resolution, logging, code compares, version backups, and so on.

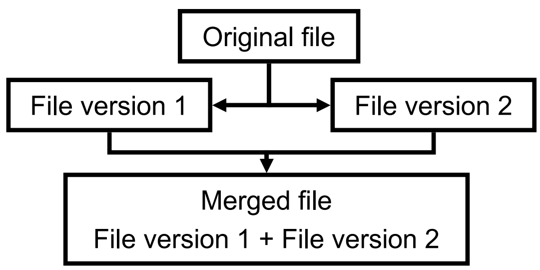
ORACLE

Oracle BI Repository Development Process

Developers can work simultaneously on repository development, managing the process with utilities and interfaces in the Administration Tool.

SCM Three-Way Merge Process

- Manages concurrent development
  - Highly restrictive alternative is serial development.
- Permits changes to the same file by multiple developers
- Requires merging and reconciliation:
  - Most merging is automatic; changes generally do not conflict.
  - Conflicts require manual intervention.
- Creates a fourth “merged” file based on two changed files, which are base-lined against a common parent file



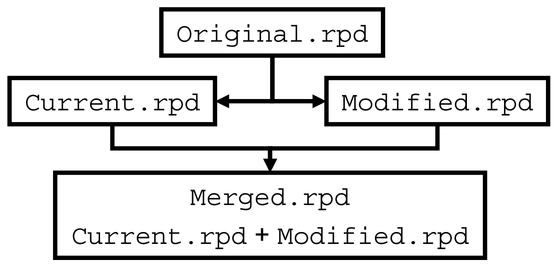
SCM Three-Way Merge Process

The classic Software Configuration Management (SCM) process uses a three-way merge to manage concurrent development. This permits changes to the same file by multiple developers. Changes are managed by merge and reconciliation.

Oracle BI Repository Three-Way Merge Process

Conceptually identical to classic SCM three-way merge:

- Oracle BI repository is stored as a file (`.rpd`).
- Merge is managed by using the Administration Tool.



Oracle BI Repository Three-Way Merge Process

The Oracle BI repository is stored as a file. Developers check out the file and make changes locally. Changes are then merged into a final, merged repository file.

Multiuser Development Projects

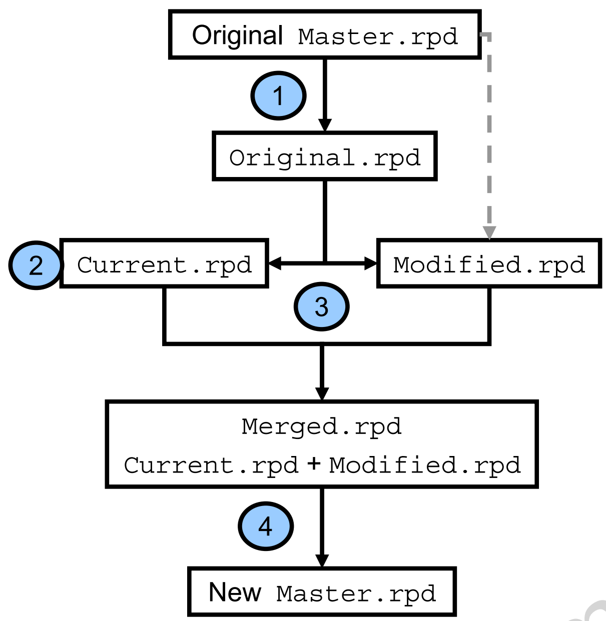
- Projects:
  - Are subsets of repository metadata
  - Consist of Presentation layer subject areas and their associated logical facts, dimensions, application roles, users, variables, and initialization blocks
  - Can overlap with other projects
- The recommended practice is to create projects of manageable size based on individual logical star schemas in the business model.

ORACLE

Overview: Oracle BI Multiuser Development

The developer:

1. Checks out projects from the master repository
2. Makes changes in the local (current) repository
3. Merges the local changes
4. Publishes to the network



ORACLE

Overview: Oracle BI Multiuser Development

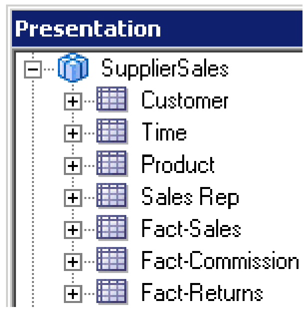
The Oracle BI multiuser development process follows a purposeful three-way merge. The developer performs the following steps:

1. Check out projects from the master repository, which is stored in the shared multiuser directory. An unalterable copy of the checked out repository (`original.rpd`) is automatically retained by the system for use during the merge.
2. Make changes in the local (current) version of the repository. The modified repository contains changes by other developers between checkout and merge.
3. Merge the local changes. The original master repository may have changed through concurrent development since checkout. A copy of the latest master repository (modified) is automatically retrieved by the system and compared with the current and original repositories in a three-way merge. The modified master repository is automatically locked by the system to prevent issues during merge. If there are any configuration conflicts during the merger, the developer resolves them manually.
4. Publish the new master repository to the network. The system automatically moves the merged repository to the shared multiuser directory and removes the locks. The merged repository is the new master repository.

This slide provides an overview of the process. Each step is covered in detail later in this lesson.

ABC Example

ABC wants multiple developers to be able to modify objects in the `SupplierSales` subject area simultaneously.



Steps to Set Up an Oracle BI MUDE

1. Create projects.
2. Edit projects.
3. Set up a shared network directory.
4. Copy the master repository to the shared directory.

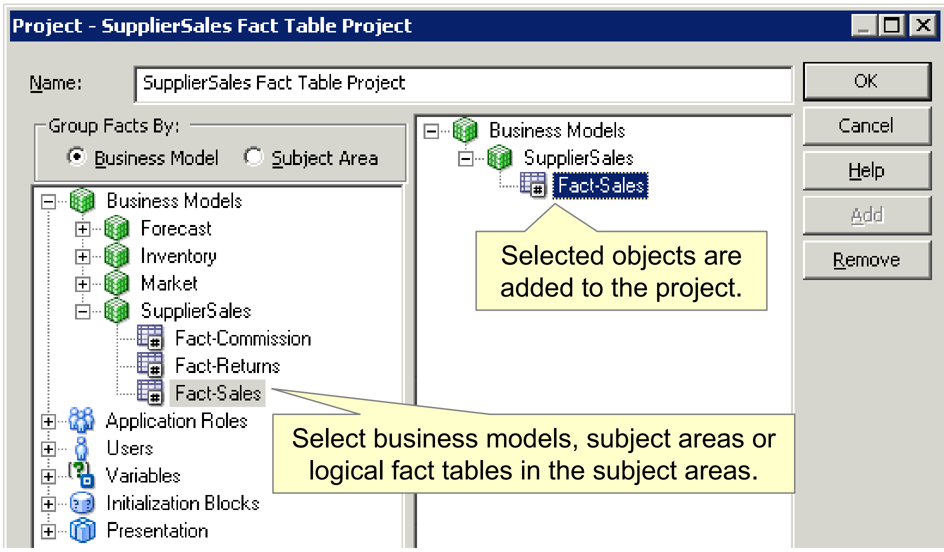
ORACLE

Steps to Set Up an Oracle BI MUDE

This slide identifies the high-level steps for setting up an Oracle BI multiuser development environment (MUDE). Each step is covered in detail in the following slides.

1. Create Projects

Select Manage > Projects to open the Project Manager.
Then select Action > New Project.



ORACLE

1. Create Projects

To create projects in the Administration Tool, select Manage > Projects to open the Project Manager. Then select Action > New Project. The left pane contains the objects that are available to be placed in a project. The right pane contains the objects that you select to be part of the project.

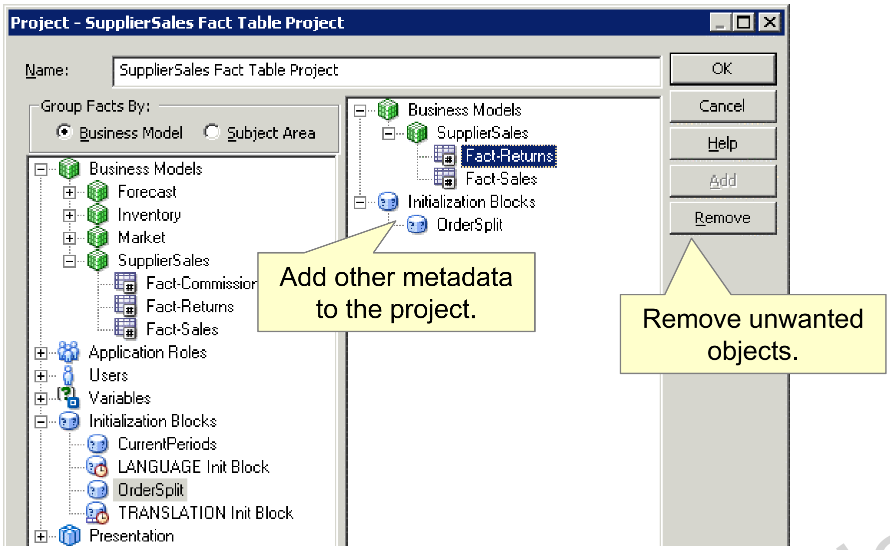
Enter a name for the project. Build the project by adding subject areas or logical fact tables to the project. You can group facts by subject area or by business model. You can select one or more logical fact tables in the business model that are related to the subject area and then click Add. Or you can select a subject area and then click Add. The Administration Tool adds all the logical fact tables automatically.

Adding a subject area includes all fact tables and dependencies in the subject area. Adding a logical fact table includes the subject area containing the table. In both cases, logical dimension tables joined to the logical fact tables are implicitly included, even though they do not appear in the right pane.

In the example in the slide, a new project called `SupplierSales` Project is created. The `SupplierSales` subject area and the related fact table, `Fact-Sales`, are added to the project.

2. Edit Projects

- Remove unwanted objects from the project.
- Add other metadata (such as users, initialization blocks, or variables) to the project.



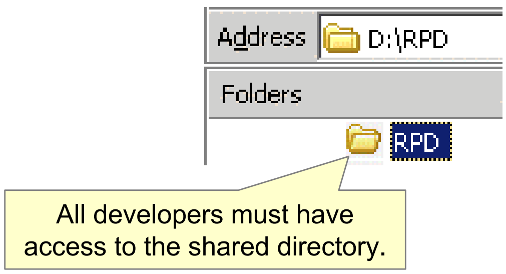
2. Edit Projects

To remove objects from the project, select the object in the right pane and then click Remove.

To add additional metadata objects, select the object in the left pane and click Add, or double-click the object in the left pane. Add additional subject areas, application roles, users, variables, or initialization blocks needed for the project.

3. Set Up a Shared Network Directory

The administrator must identify or create a shared network directory that all developers can access.



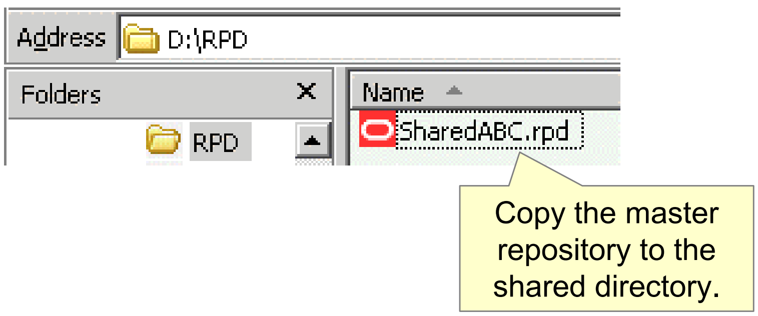
3. Set Up a Shared Network Directory

The administrator must identify or create a shared network directory that all developers can access and then copy the appropriate repository files to that location. This shared network directory is used only for multiuser development for the master repository. This directory typically contains copies of master repositories that multiple developers access during check in and check out. Developers create a pointer to this directory when they set up the Administration Tool on their machines. This directory must be accessible to all developers and repository servers.

Note: The administrator must set up a separate, shared network directory that is dedicated to multiuser development. If it is not set up and used as specified, critical repository files can be unintentionally overwritten and repository data can be lost. In the practices for this lesson, the shared directory is a local directory, `D:\RPD`. This is the example used in the remaining slides.

4. Copy the Master Repository to the Shared Directory

Copy the master repository file and paste it in the directory that you have dedicated to multiuser development.



4. Copy the Master Repository to the Shared Directory

Copy the master repository file and paste it in the directory that you have dedicated to multiuser development. Projects from this master repository are extracted and downloaded by the developers, who make changes and then merge them back into the master repository. After you copy the repository to the shared network directory, you can notify developers that the multiuser development environment is ready for use.

Making Changes in an Oracle BI MUDE

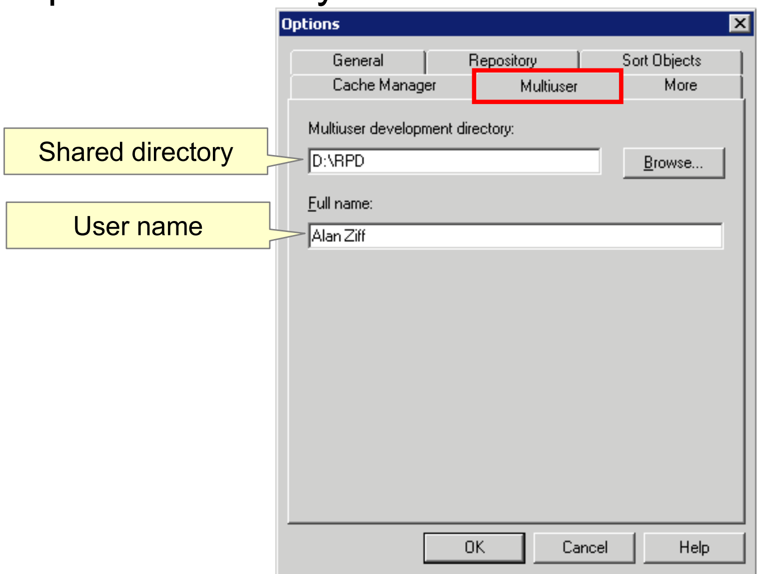
1. Point to the multiuser directory.
2. Check out projects.
3. Tasks performed by Administration Tool during checkout
4. Change metadata.
5. Multiuser options during development
6. Merge local changes.
7. Make merge decisions.
8. Publish to network.
9. Track project history.

Making Changes in an Oracle BI MUDE

This slide identifies the high-level steps and processes that occur when you make changes in an Oracle BI multiuser development environment. Each step is covered in detail in the following slides.

1. Point to the Multiuser Directory

Before checking out projects, each developer must set up the Administration Tool application to point to the multiuser development directory.



1. Point to the Multiuser Directory

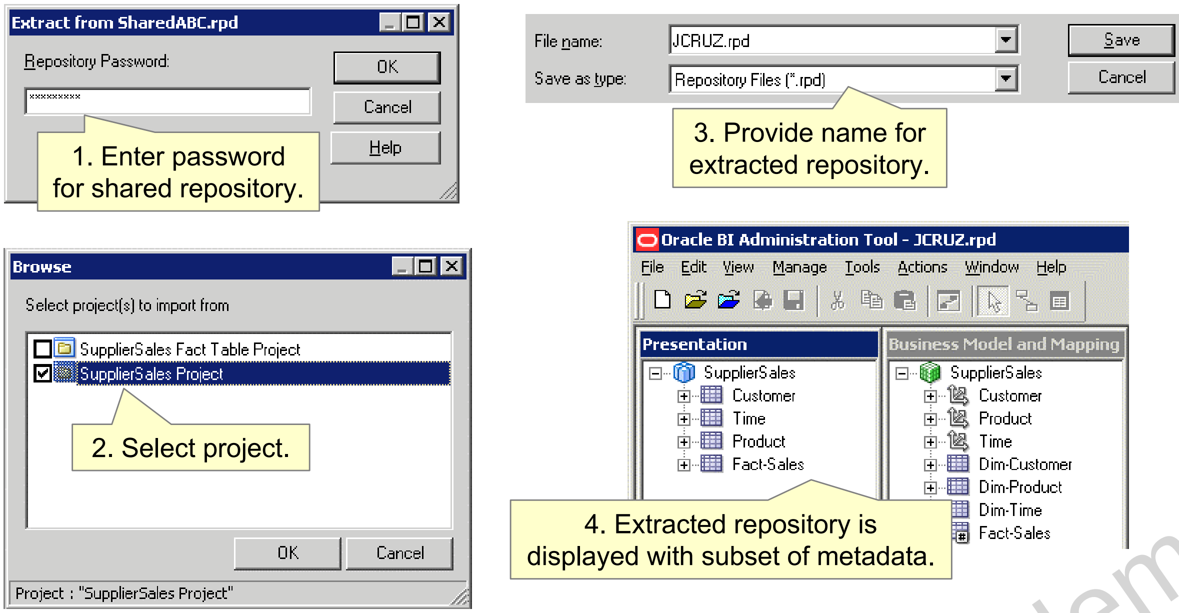
Select Tools > Options and then click the Multiuser tab.

The “Multiuser development directory” field is mandatory. It must be completed by any user who wants to use the multiuser development (MUD) feature and must be set to the directory on the network shared with other MUD developers. Use the Browse button to navigate to the directory or enter the directory path. The Administration Tool stores this path in a hidden Windows registry setting on the developer’s workstation and uses it during checkout and check in.

The “Full name” field is optional. If a user enters a name here, the value is used by default in the “Full name” field of the repository Lock Information dialog box (discussed later). For convenience and tracking, each MUD developer should enter a full name. The value is stored in the `HKEY_CURRENT_USER` part of the registry and is, therefore, unique for each login.

2. Check Out Projects

Select File > Multiuser > Checkout and select the desired project or projects.



ORACLE

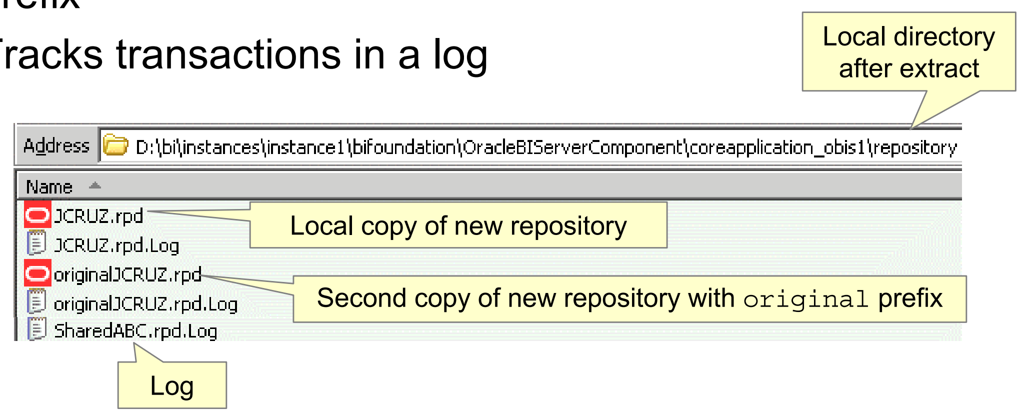
2. Check Out Projects

After setting up a pointer to the multiuser development default directory, a developer can check out the desired projects.

- To check out projects, select File > Multiuser > Checkout. The Checkout option is available only when there is a multiuser development directory defined on the Multiuser tab of the Options dialog box.
- The developer is presented with a dialog box asking for the repository password.
- After entering the password, the developer is presented with a dialog box to select the project or projects to import. If there is only one project in the master repository, it is chosen by default and no dialog box is presented to the user. In this example, there are two projects.
- After selecting a project or projects, the user must enter the name of the new, extracted repository, which is stored in the user's local directory.
- The extracted repository is displayed with the subset of metadata from the `SupplierSales` Project. Note that it contains only the `SupplierSales` subject area and business model. The other ABC subject areas and business models are not shown. Note also that the `SupplierSales` subject area and business model contain only the `Fact-Sales` fact table.

3. Administration Tool Tasks During Checkout

- Saves a local copy of the selected projects in a new repository in the local directory
- Saves a second local copy of projects in the new repository in the local directory with `original` as the prefix
- Tracks transactions in a log



3. Administration Tool Tasks During Checkout

During checkout, the Administration Tool performs the following tasks:

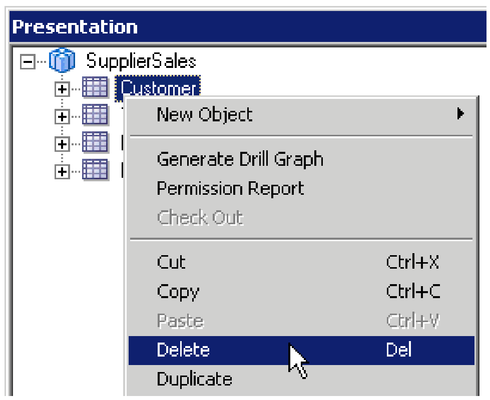
- In the developer's local `\coreapplication_obis1\repository` directory, the Administration Tool saves a local copy of the selected projects in a new repository. In this example, the new repository is named `JCRUZ.rpd`. The developer makes metadata changes in this file.
- In the developer's local `\coreapplication_obis1\repository` directory, the Administration Tool saves a second local copy of the new repository, adding `original` as the prefix, to enable changed projects to be compared with original projects locally. In this example, the local copy is named `originalJCRUZ.rpd`.

All changes are tracked in the log, which is `SharedABC.rpd.Log` in this example.

Caution: When a developer selects and saves projects to a local repository file, the Administration Tool does not place a lock on the projects in the master repository on the shared network drive. Therefore, nothing physically prevents others from working on the same project. To determine if a project has been checked out, you need to check the multiuser development log in the log viewer (discussed later in this lesson).

4. Change Metadata

Change metadata as you would during single-user development, with the exception of physical connection settings.



4. Change Metadata

Most changes that can be made to standard repository files are also supported for local repository files. Developers can add new logical columns, logical tables, change table definitions, logical table sources, and so on. In this example, the `Customer` presentation table is deleted. Developers may also work simultaneously on the same project locally. It is important to note, however, that Oracle BI assumes that the individual developer understands the implications that these changes might have on the master repository. For example, if a developer deletes an object in a local repository, this change is propagated to the master repository without a warning prompt.

Physical connection settings should not be modified in a local repository. To prevent developers from overwriting passwords and other important objects in the master repository, the physical connection settings, security settings, and database feature table changes are not retained in a multiuser development merge.

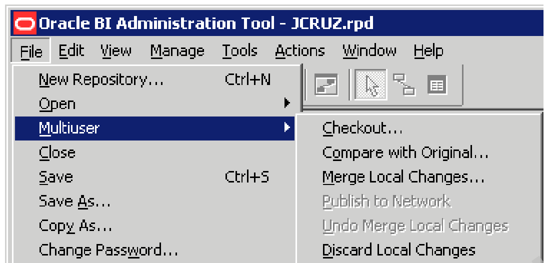
After making changes to a local repository, the developer can load the modified repository and test the edited metadata.

Note: All repository connection information specified in the repository metadata (ODBC, OCI, and so on) must also exist on the developer's workstation.

5. Multiuser Options During Development

The following multiuser options are enabled when the local, extracted repository is open:

- Compare with Original
  - Launches a dialog box that compares the local version of the original repository with the subset repository
- Discard Local Changes
  - Discards changes to the local repository without checking in
- Merge Local Changes
  - Launches a dialog box to merge local changes with the master repository

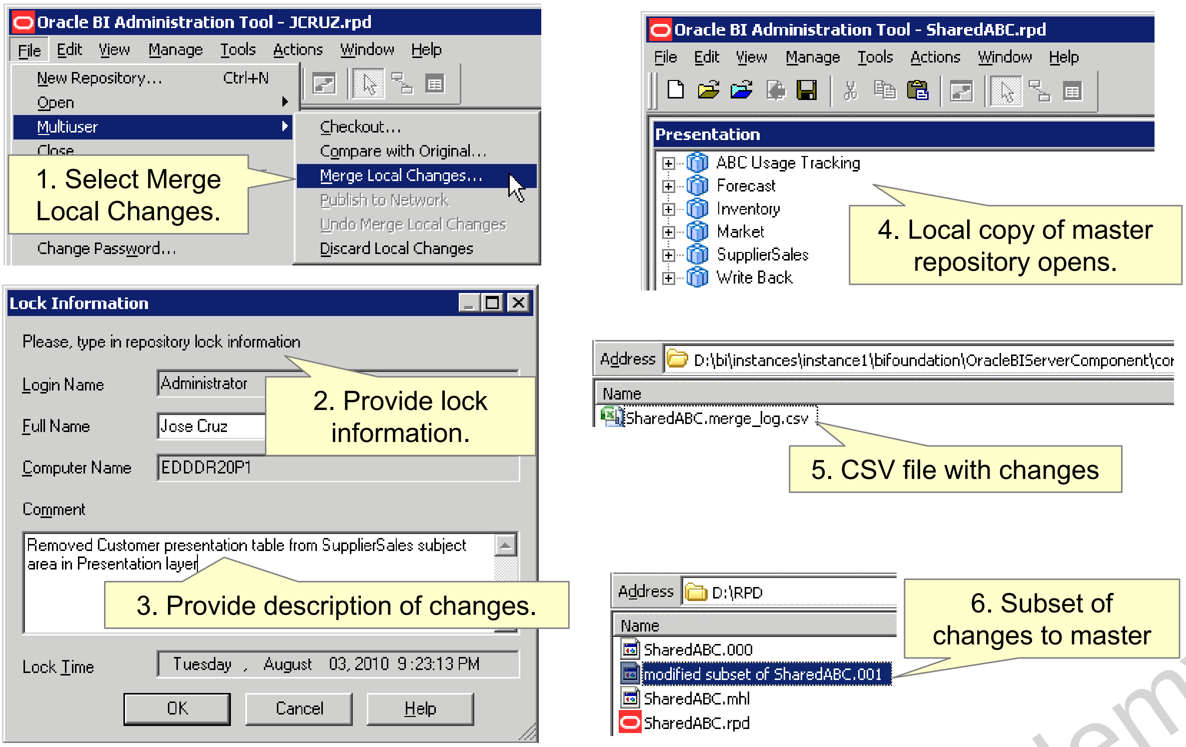


5. Multiuser Options During Development

When a developer opens the local version of the extracted repository, the following multiuser options are enabled:

- **Compare with Original** launches a compare repositories dialog box, which compares the local version of original repository with the subset repository.
- **Discard Local Changes** closes the repository and discards any changes to the local repository without checking in changes.
- **Merge Local Changes** first presents a Lock Information dialog box to lock the master repository in the shared directory, and then opens the standard Merge dialog box.

6. Merge Local Changes

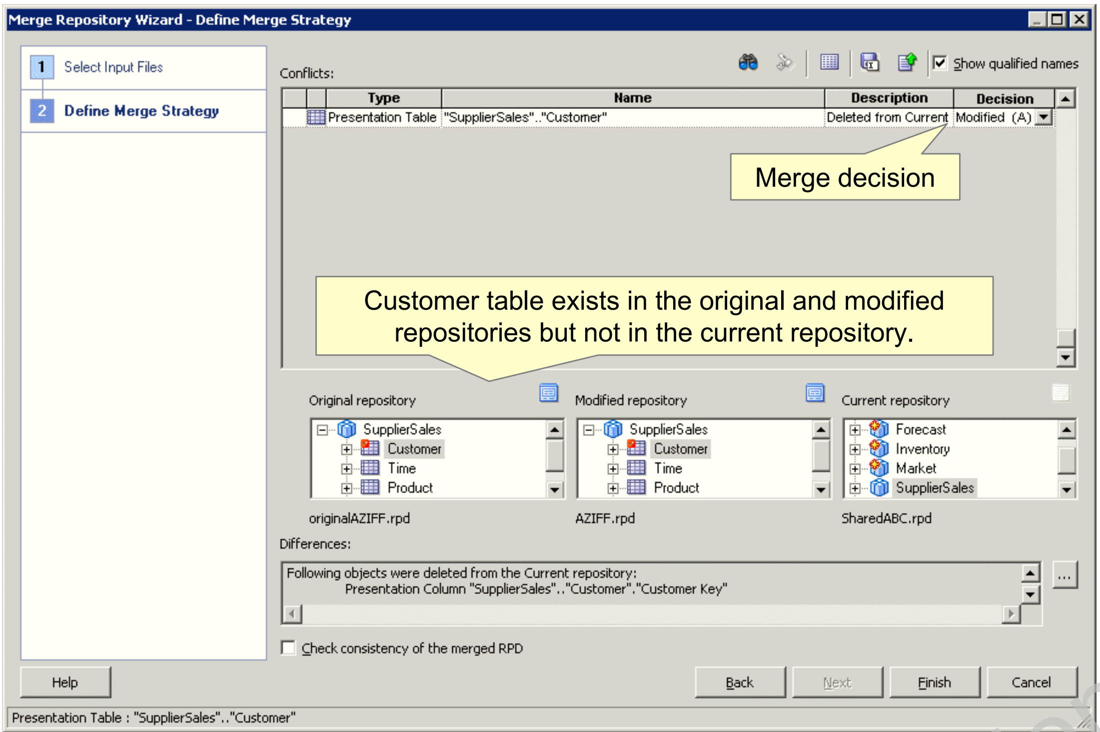


6. Merge Local Changes

After making changes to the local repository, the developer begins the check-in process by selecting Multiuser > Merge Local Changes. When the check-in process begins, the following actions occur:

- The Administration Tool determines whether the master repository is currently locked. If not, it locks the master repository (preventing other developers from performing a merge until the current merge is complete) and records the lock in the log. For other developers, the Multiuser Development > Merge Local Changes option on the File menu is unavailable until the current check-in process has been successfully completed.
- If there had been any conflicts, the Merge Repository Wizard would have opened and displayed the Define Merge Strategy window. You would then make merge decisions about whether to include or exclude objects by choosing Current or Modified from the Decision list. You learn more about this process later in this lesson.
- A local version of the master repository opens.
- The Administration Tool automatically creates a comma-separated value file in the local repository directory on the developer's machine. This file lists changes made to the master repository during the merge.
- A new file is created in the shared master repository directory with a modified subset of the master repository.

7. Make Merge Decisions



7. Make Merge Decisions

If there are conflicts when a developer selects File > Multiuser > Merge Local Changes, the developer has to make decisions about what to do with the changes.

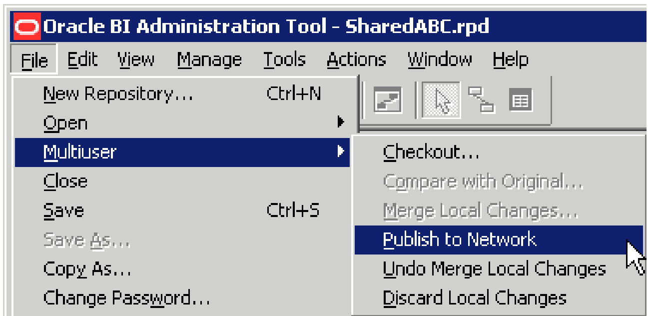
If there are conflicts, the Merge Repository Wizard opens and compares the original repository, the modified repository, and the local version of the current shared repository, which was opened when the developer selected Merge Local Changes. In this example, the original and modified repositories have a `Customer` presentation table that is not in the current shared repository. This means that the first developer made this change and checked in the project while the second developer had the project checked out. At this point, a decision must be made about how to proceed with the merge.

The dialog box gives the developer the option of accepting or not accepting the changes in the master repository. Selecting Modified means that the developer retains the changes in the modified repository and overwrites other developers' work in the current repository. Selecting Current means that the developer accepts the changes by other developers in the current local version of the master repository. The developer must select either Current or Modified. Changes cannot be combined. Thus, if two developers modify the same object at the same time, one developer's changes are lost.

In this example, the second developer selects Modified, which means that the `Customer` table is added back to the master repository.

8. Publish to Network

Select File > Multiuser > “Publish to Network” to commit changes to the master repository.

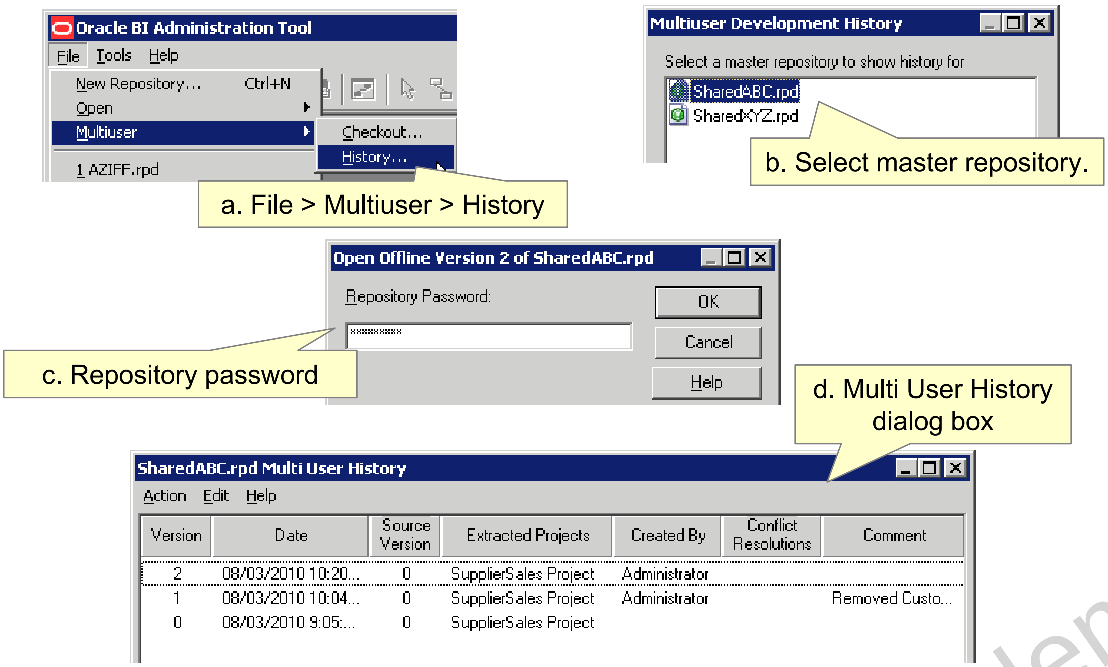


8. Publish to Network.

The next step is to commit changes to the master repository. To commit changes to the master repository in the multiuser development directory, select File > Multiuser > “Publish to Network,” and then click OK. The local copy of the master repository closes and the master repository in the multiuser development directory is overwritten with the copy of the repository containing the developer's changes. Notice that you also have the option to discard local changes or undo merge at this point.

9. Track Project History

Project history is stored in a log viewer.



9. Track Project History

All project log events are stored in a log viewer:

a. The viewer is accessed by selecting File > Multiuser > History. This menu item is available only when the Oracle BI Administration Tool is open with no repository file open.

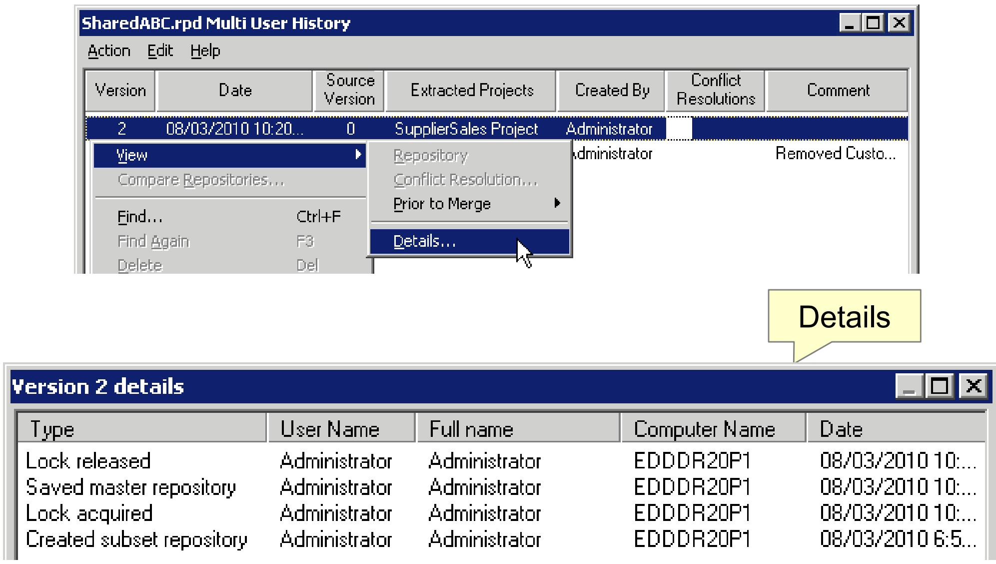
b. When this menu item is selected, the user sees the Multiuser Development History dialog box. This dialog box lists all master repositories in the shared Multiuser Development Directory specified in the Options dialog box. If no directory is specified in the Options dialog box, the History menu item is disabled. If the directory contained only one master repository, it would be selected by default and no Multiuser Development History dialog box would be presented to the user.

c. After the repository has been selected, the user is prompted to fill in the user ID and the password for the latest version of the master repository.

d. After successful login, the Multi User History dialog box is displayed with the different versions of the projects listed.

History Menu Options

Use menu options to navigate and view history.

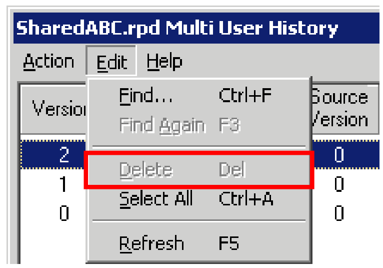


History Menu Options

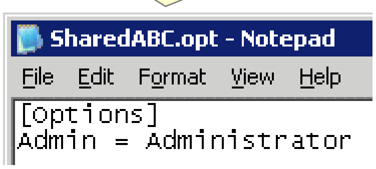
- **View > Repository** loads the selected master version of the repository to the Administration Tool in read-only mode.
- **View > Prior to Merge > Projects** loads the selected version of the modified subset repository to the Administration Tool in read-only mode.
- **View > Prior to Merge > Changes** compares the modified subset repository of a selected version with the original subset repository. It opens the modified subset of the shared repository and displays the Compare Repositories dialog box with all changes made by the user in the selected version.
- **View > Details** displays the detail log for the selected version or multiple selected versions, or all details if no version is selected. The slide shows an example.
- **View > Conflict Resolution** loads all necessary repositories of the selected version and shows the Merge dialog box in read-only mode with all selected decisions as they were during the Merge Local Changes activity at that time. The Conflict Resolution check box must be selected in the dialog box for this menu item to be enabled. Otherwise, there is nothing to show because there were no decisions made by the user.

Deleting History Items

The Delete menu item is available only to administrators who are defined in a hidden option file in the MUD directory.



Hidden option file



Stored in MUD directory

Deleting History Items

The Delete menu item is available only to an administrator. Administrators are defined in a special hidden option file in the MUD directory. The file has to have a hidden flag. The file can have network access privileges set to be accessed only by the MUD administrator(s). The file must have the same base name as the master repository, but the extension is `.opt`. For example, for `\network\RPD\SharedABC.rpd`, the administrator can create the hidden file named `\network\RPD\SharedABC.opt`.

The option file is a normal text file in the following format:

```
[Options]
Admin=admin1;admin2
```

Administrators are defined by their network login name. There may be more than one administrator. In this case, administrator names are separated by semicolons.

Example:

```
[Options]
Admin=Administrator;MWEST;JMEYER
```

An administrator can delete the whole MUD history or the oldest 1 to *n* versions. It is not possible to delete version(s) in the middle of the history. For example, an administrator cannot delete version 3 if there are versions 2 and 1. If an administrator deletes the entire MUD history, version counting restarts from version 1. If an administrator leaves one or more versions in the history, the version number remains the same as it was last time.

Summary

In this lesson, you should have learned how to:

- Set up an Oracle BI multiuser development environment
- Describe multiuser development environment functionality
- Develop a repository with multiple developers

Practice 22-1 Overview: Setting Up a Multiuser Development Environment

This practice covers the following topics:

- Creating projects
- Copying a master repository to a shared directory
- Setting a multiuser shared directory

Practice 22-1 Overview: Setting Up a Multiuser Development Environment

ABC is accustomed to using multiuser development environments for its developers. You prepare the development platform to support multiuser development and then configure two users to act as developers to test the environment.

Practice 22-2 Overview: Using a Multiuser Development Environment

This practice covers the following topics:

- Checking out projects
- Modifying project metadata
- Checking in projects
- Publishing changes to the network
- Merging changes
- Checking project history

Practice 22-2 Overview: Using a Multiuser Development Environment

In this practice, two developers, `JCRUZ` and `AZIFF`, work in the Oracle BI multiuser development environment and modify the same project simultaneously, including checking out and checking in projects and merging metadata. This requires you to have two instances of the Oracle BI Administration Tool running at the same time.

23

Configuring Write Back in Analyses

Objectives

After completing this lesson, you should be able to configure write back in analyses.

Write Back in Analyses

Provides users of a dashboard page or an analysis with the ability to modify the data that they see in a table view

DOLLARS column is enabled for write back.

TYPECODE	ITEMTYPE	MONTHCODE	DOLLARS
100	Baking	200801	271234
		200802	289782
		200803	316107
		200804	298547
		200805	336265

Write Back in Analyses

Users of a dashboard page or an analysis have the ability to modify the data that they see in a table view. This ability is often referred to as *write back*. In the example in the slide, the `DOLLARS` column is enabled for write back.

As a repository developer, you assist the system administrator and content designer in configuring write back for users.

Steps to Configure Write Back

1. Create a physical table with write back columns.
2. Import the write back table.
3. Enable write back for the connection pool.
4. Enable write back for logical columns.
5. Set write back permissions in the Presentation layer.
6. Enable write back in `instanceconfig.xml`.
7. Create a write back template.
8. Store the write back template.
9. Grant write back privileges.
10. Create an analysis with columns enabled for write back.
11. Override the default data format.
12. Enable write back in the table view.
13. Verify results.

1. Create a Physical Table with Write Back Columns

- Create a physical table in the database that has a column for each write back field needed.
- For optimum security, store write back database tables in a unique database instance.

```
CREATE TABLE "SUPPLIER2"."D1_FORECAST"
   (     "TYPECODE" NUMBER(5,0),
         "MONTHCODE" NUMBER(10,0),
         "DOLLARS" NUMBER(10,0),
         "UNITORDD" NUMBER(10,0)
   )
```

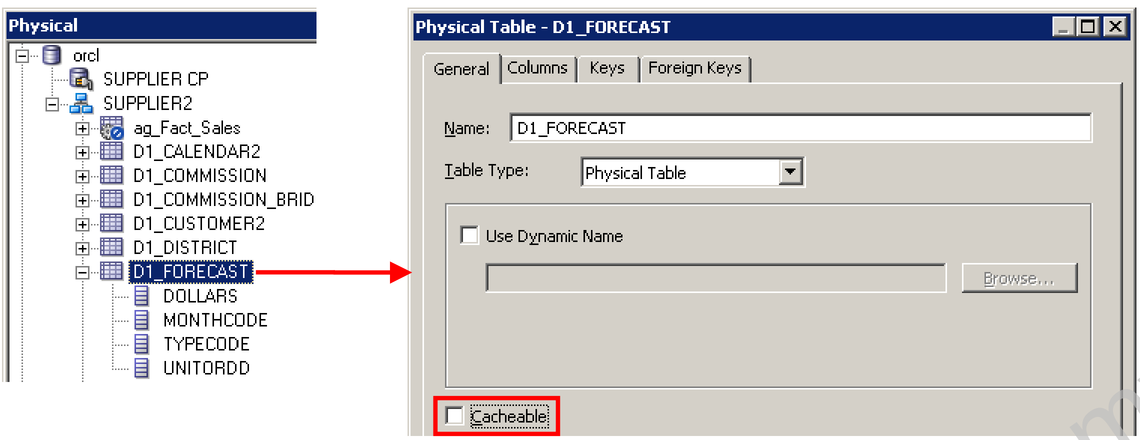
TYPECODE	MONTHCODE	DOLLARS	UNITORDD
120	200809	308373	20018
120	200810	373023	23848
120	200811	308374	19592
120	200812	334670	21267
120	200901	338604	21357
120	200902	313858	20232
120	200903	329071	22021
120	200904	212220	13875

1. Create a Physical Table with Write Back Columns

In this example, you create a `D1_FORECAST` table in the `SUPPLIER2` schema. You then configure the `DOLLARS` column to be the write back column.

2. Import the Write Back Table

- Import the write back table into the Physical layer of the repository.
- Disable the Make Table Cacheable property for the physical table.



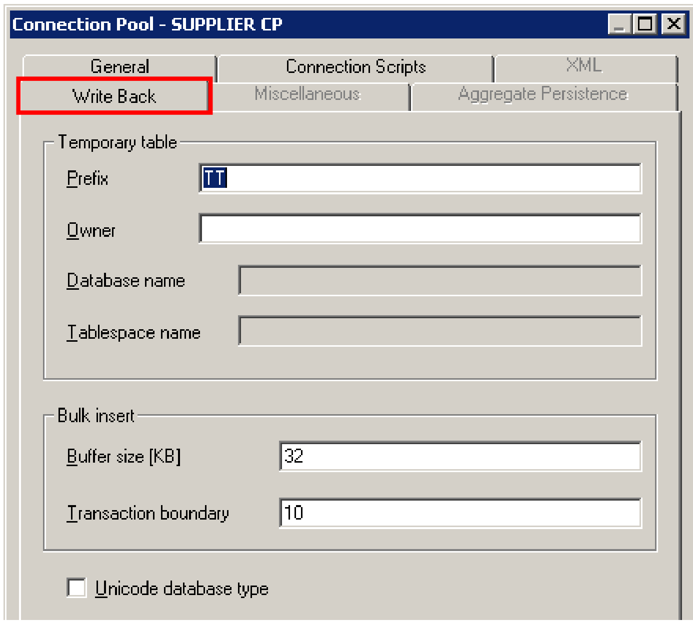
2. Import Write Back Table

Using the Administration Tool, import the write back table into the Physical layer of the repository. Disable the Cacheable property for the physical table. This ensures that data written back to the database is displayed to the user and is not a cached value.

In this example, the Cacheable object is deselected for the `D1_FORECAST` table in the Physical layer of the repository.

3. Enable Write Back for the Connection Pool

Set properties on the Write Back tab in the Connection Pool dialog box.



3. Enable Write Back for the Connection Pool

In the Physical layer, double-click the connection pool to open the Connection Pool dialog box. Click the Write Back tab. If you are using an Oracle data source, you can accept the defaults, which are shown in the screenshot: `TT` is selected for the temporary table prefix, buffer size = `32`, and transaction boundary = `10`.

Prefix: When the Oracle BI Server creates a temporary table, these are the first two characters in the temporary table name. The default value is `TT`.

Owner: Table owner name that is used to qualify a temporary table name in a SQL statement (for example, to create the `owner.tablename` table). If this field is blank, the user name specified in the writeable connection pool is used to qualify the table name, and the Shared Logon field on the General tab should also be set.

Database name: Database where the temporary table is created. This property applies only to IBM OS/390 because IBM OS/390 requires the database name qualifier to be part of the `CREATE TABLE` statement.

3. Enable Write Back for the Connection Pool (continued)

Tablespace name: Tablespace where the temporary table is created. This property applies only to OS/390 because OS/390 requires the tablespace name qualifier to be part of the `CREATE TABLE` statement.

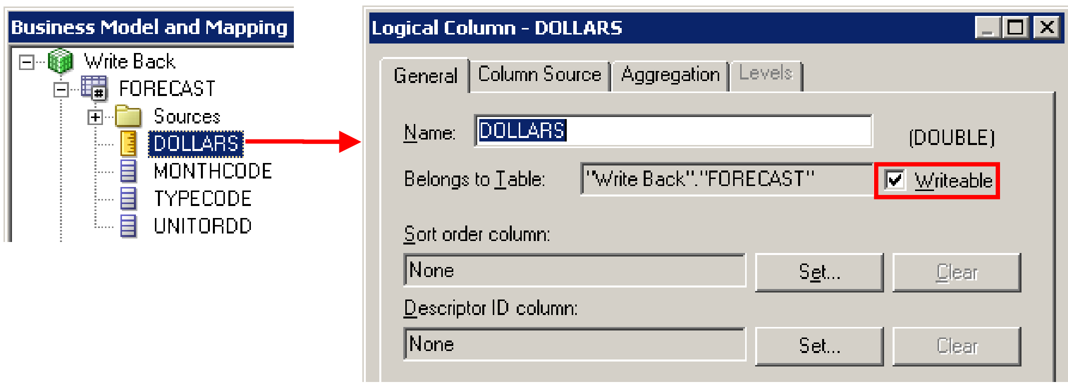
Buffer size (KB): Used for limiting the number of bytes each time data is inserted in a database table. For optimum performance, consider setting this parameter to `128`.

Transaction boundary: Controls the batch size for an insert in a database table. For optimum performance, consider setting this parameter to `1000`.

Unicode database type: Select this option when working with columns of an explicit Unicode data type (such as `NCHAR`) in a Unicode database. This ensures that the binding is correct and data is inserted correctly.

4. Enable Write Back for Logical Columns

In the BMM layer, enable write back for logical columns that will be used for write back in an analysis.



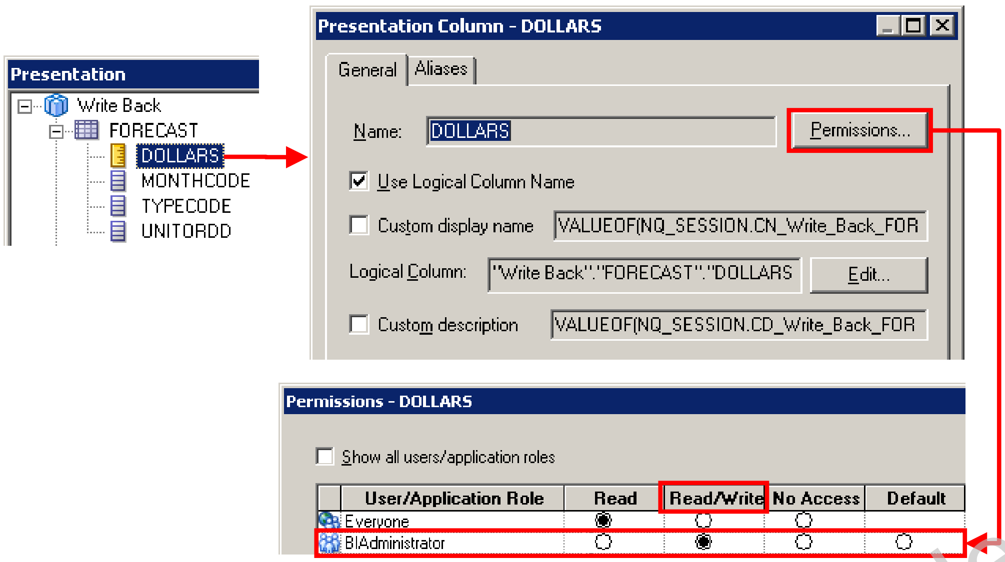
4. Enable Write Back for Logical Columns

Open the Logical Column dialog box and select Writeable on the General tab.

In this example, you are enabling write back for the `DOLLARS` column in the `FORECAST` logical table in the Write Back business model.

5. Set Write Back Permissions in the Presentation Layer

Select the Read/Write permission for any application roles or users who should have write back permission for an object in the Presentation layer.

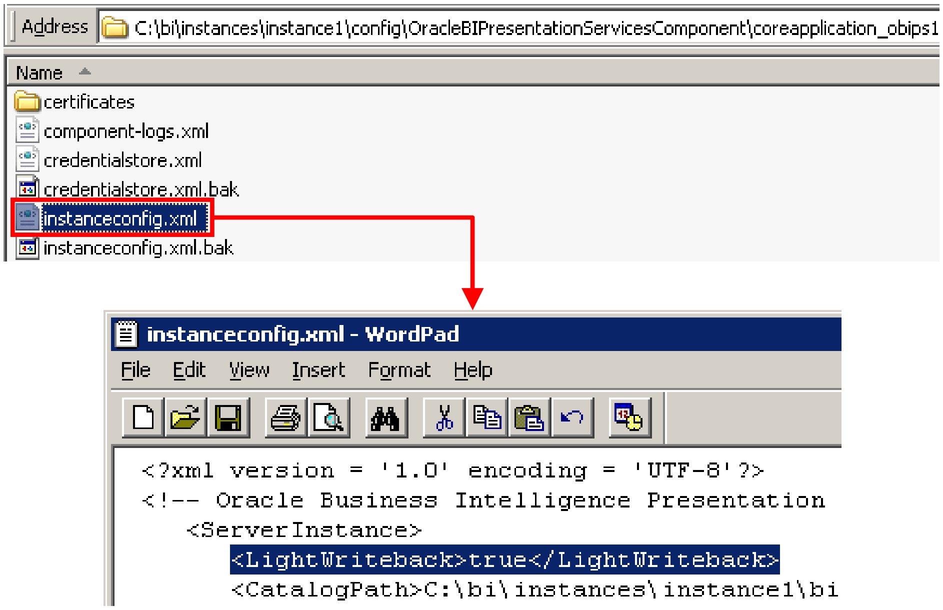


5. Set Write Back Permissions in the Presentation Layer

In this example, you give Read/Write permission to the BIAdministrator application role for the `DOLLARS` presentation column.

6. Enable Write Back in `instanceconfig.xml`

Set the `LightWriteback` parameter to `true`.



6. Enable Write Back in `instanceconfig.xml`

Navigate to the following location:

<ORACLE_INSTANCE>\config\OracleBIPresentationServicesComponent\coreapplication_obips1

Open `instanceconfig.xml` in an editor.

In the `ServiceInstance` section, set the `LightWriteback` parameter to `true` by entering `<LightWriteback>true</LightWriteback>`.

7. Create a Write Back XML Template

The write back template is an XML-formatted file that contains SQL statements that are needed to insert and update records in the write back table and columns that you have created.

```
  <?xml version="1.0" encoding="utf-8" ?>
- <WebMessageTables xmlns:sawm="com.siebel.analytics.web/message/v1">
  - <WebMessageTable lang="en-us" system="WriteBack" table="Messages">
    - <WebMessage name="SetForecast">
      - <XML>
        - <writeBack connectionPool="SUPPLIER CP">
            <insert>INSERT INTO D1_FORECAST (TYPECODE,
              MONTHCODE, DOLLARS)VALUES (@1,@3,@4)</insert>
            <update>UPDATE D1_FORECAST SET DOLLARS=@4
              WHERE TYPECODE=@1 AND
              MONTHCODE=@3</update>
          </writeBack>
        </XML>
      </WebMessage>
    </WebMessageTable>
  </WebMessageTables>
```

ORACLE

7. Create a Write Back Template

You can create multiple write back templates, customizing each one for the fields used in each specific analysis. In the analysis properties, you specify the name of the write back template to use. If a user has the "Write Back to Database" privilege, the write back fields in their analyses appear as editable fields. If the user does not have this privilege, the write back fields appear as normal fields.

Requirements for the Write Back Template

Notice that the `WebMessage` name element is `SetForecast`. To ensure that write back works correctly, in the `WebMessage` element of the file you must include the name of the SQL template that you will specify when you create the write back table (described later in this lesson). The name is arbitrary.

To meet security requirements, you must specify the connection pool along with the SQL commands to insert and update records. In this example. the connection pool is set to `SUPPLIER CP`.

7. Create a Write Back Template (continued)

The SQL commands reference the values passed in the write back schema to generate the SQL statements to modify the database table. Values can be referenced either by position (such as `@1`, `@3`) or by column ID (`@{c0}, @{c2}`). Column positions start numbering with `1`, whereas column IDs start with `c0`. In this example, values will be inserted or updated for the `TYPECODE`, `MONTHCODE`, and `DOLLARS` columns in the `D1_FORECAST` table.

You can include `<insert>` and `<update>` elements in the template. The insert tag is necessary only if there are null values in the write back physical column. This example shows both elements. Oracle BI Server chooses between update or insert depending on whether the column is null. If you do not want to include SQL commands in the elements, you can insert a blank space between the opening and closing tags, as in the following example:

```
<insert> </insert>
```

If a parameter's data type is not an integer or real number, add single quotes around it.

8. Store the Write Back Template

Store the write back template files in the `customMessages` folder.

8. Store the Write Back Template

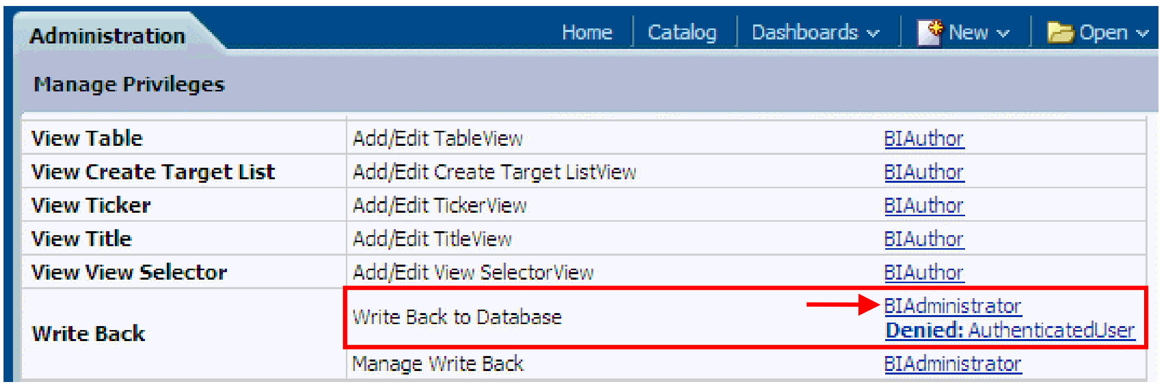
The system can have multiple `customMessages` folders, such as `ORACLE_HOME\Web\msgdb\customMessages`. This is primarily to persist these files during upgrades. Although XML message files that affect a language-specific user interface must be localized, in most cases the XML file that is used for configuring a write back template is not translated, because it is language independent.

The write back template files can have any name of your choosing, because the system reads all XML files in the `customMessages` folder. To ensure that write back works correctly, include in the `WebMessage` element of the file the name of the SQL template that you specify when you create the write back table (described later in this lesson). You can have multiple `WebMessage` elements in one file, with each element specifying one SQL template.

For more information about Web messages, refer to the *System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

9. Grant Write Back Privileges

Use the Oracle BI Presentation Services Administration screen to set the “Write Back to Database” privilege that users need for modifying values in analyses.



ORACLE

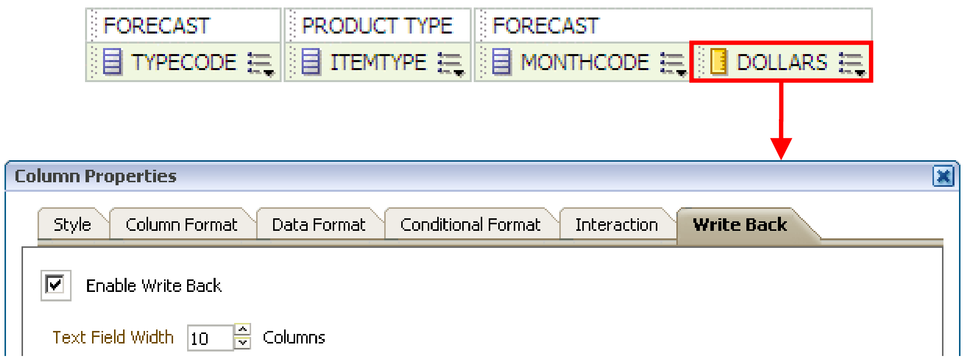
9. Grant Write Back Privileges

The “Write Back to Database” privilege enables the user interface controls for write back (editable fields and appropriate buttons) and also enables the server call that writes data back to the data source.

In this example, the “Write Back to Database” privilege is granted to the BIAdministrator application role.

10. Create an Analysis with Columns Enabled for Write Back

- Create an analysis that contains a table view with the columns configured for write back.
- For each write back column, select the Enable Write Back check box on the Write Back tab in Column Properties.



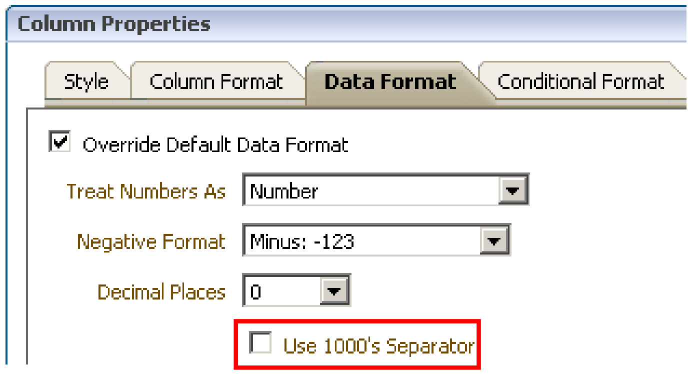
10. Create an Analysis with Columns Enabled for Write Back

For each write back column in the analysis, display the Column Properties dialog box. On the Write Back tab, select the Enable Write Back check box.

This example shows an analysis with write back enabled for the `DOLLARS` column. You can modify the text-field width for the write back column.

11. Override Default Data Format

Override the default data format on the Data Format tab to remove values that will not write back to the database.

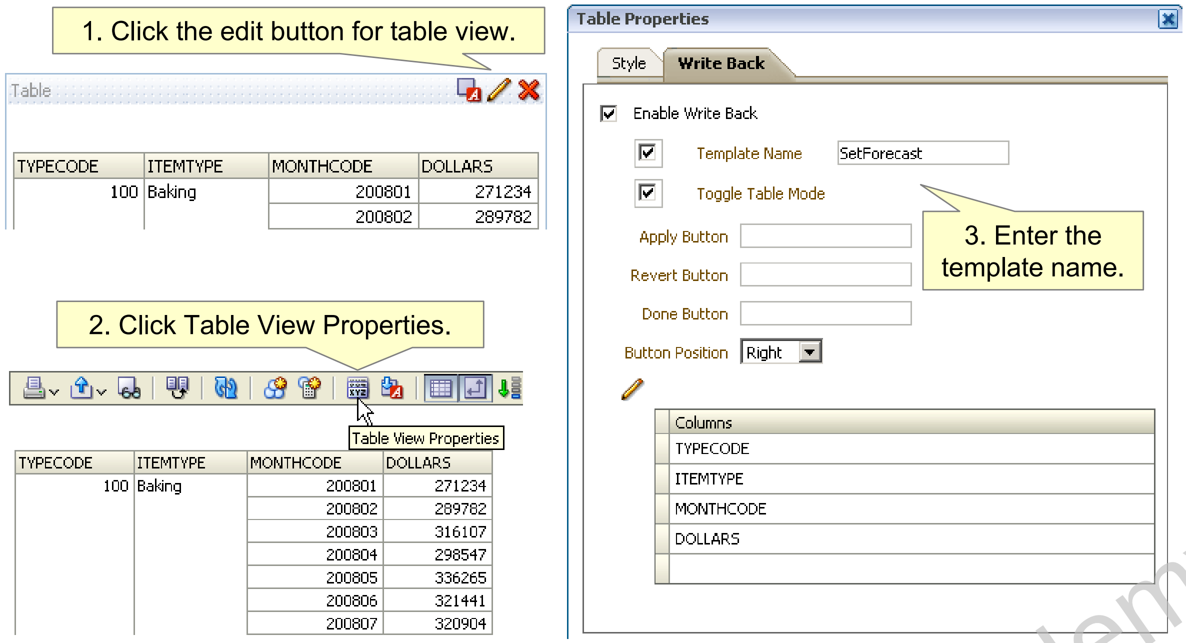


11. Override Default Data Format

In some cases, it might be necessary to override the default data format on the Data Format tab for the column that has write back enabled. For example, data does not correctly write back to the database if commas or other text values (such as currency symbols) are included in the `DOLLARS` data value.

12. Enable Write Back in the Table View

Enter the template name on the Write Back tab.



12. Enable Write Back in the Table View

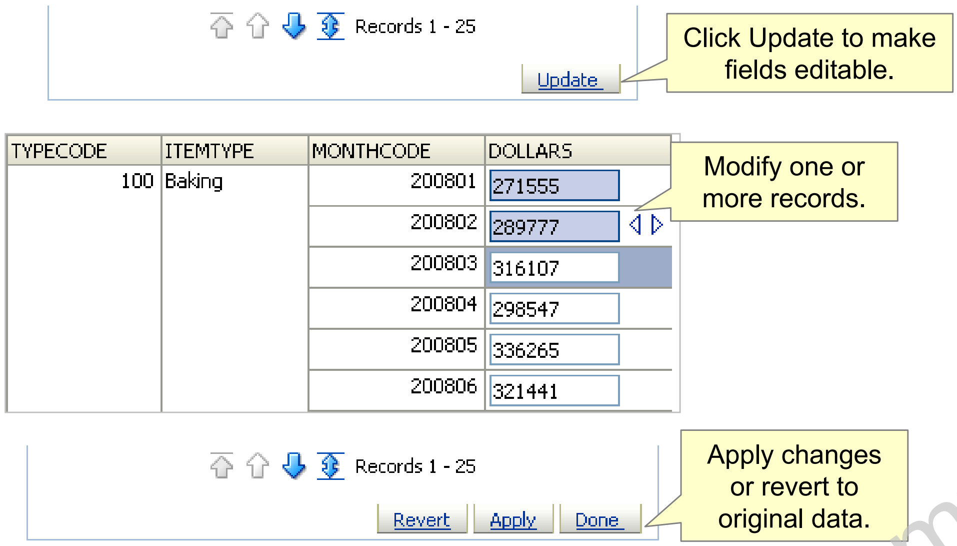
On the Results tab, click the edit button (pencil icon) for the table view. Then click the Table View Properties button to open the Table Properties dialog box.

On the Write Back tab, enter the template name. In this example, the template name is `SetForecast`. Recall that this is the Web message name in the template. Enable Write Back should be enabled by default.

At this point, you can modify the names of the Apply, Revert, and Done buttons, and you can change button positions. Toggle Table Mode enables you to switch between write back mode and the default table view.

13. Verify Results

Verify that you can enter data in the expected fields and that data is written back to the database.



13. Verify Results

Click the Update button to make the write back fields editable. Modify the data in one or more rows. Click Apply to apply the changes. Click Revert to revert to the original data. Click Done to return to the default table view.

Summary

In this lesson, you should have learned how to configure write back in analyses.

Practice 23-1 Overview: Configuring Write Back

This practice covers setting up and configuring write back for Oracle BI dashboards and analyses.

Practice 23-1 Overview: Configuring Write Back

Configuring write back involves steps that span Oracle BI components. This includes setting up write back in the repository, granting write back permissions, and enabling columns for write back in analyses.

24

Performing a Patch Merge

Objectives

After completing this lesson, you should be able to perform a repository patch merge.

Patch Merge

Provides the capability of generating an XML patch file that contains only the changes made to a repository

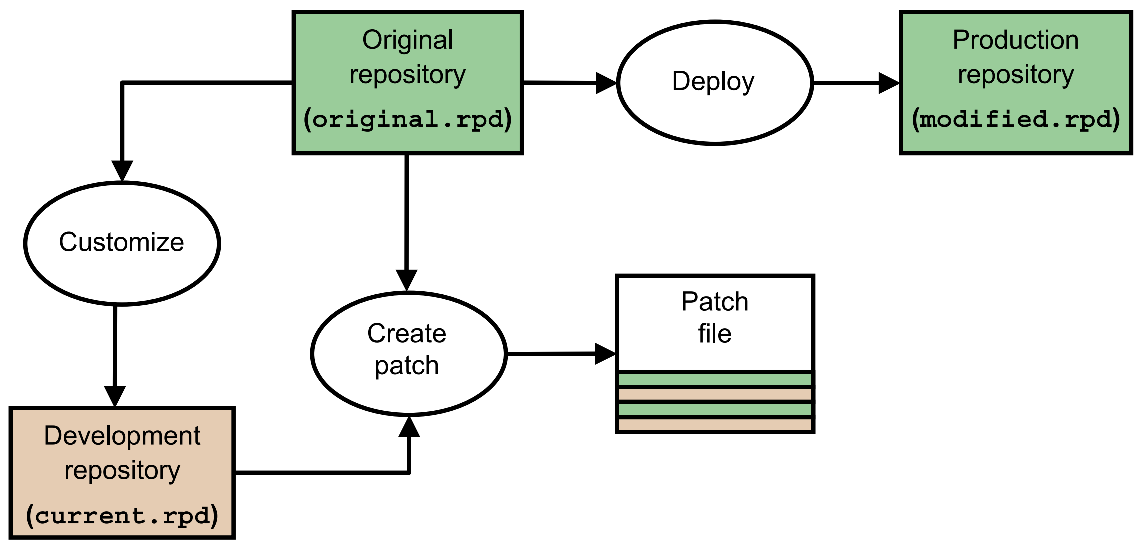
- Useful for development-to-production scenarios
- Provides a means to upgrade a repository

Patch Merge

The patch can be applied to the original version of the repository to create a new version. This is very useful for development-to-production scenarios and can also be used for Oracle BI Applications customers to upgrade their repository.

This lesson explains how to generate a patch that contains the differences between two repositories and then apply the patch to a repository file.

Creating a Patch



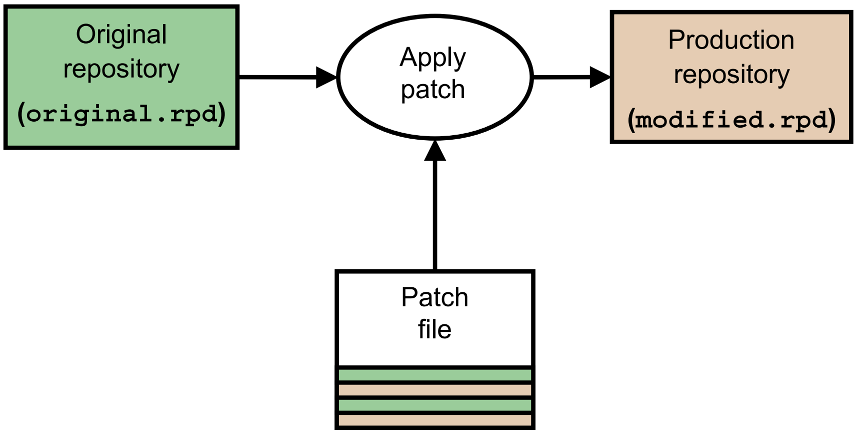
Creating a Patch

In a patch merge, you create a patch that contains the differences between the current repository file and the original repository file. Then you apply the patch file to the modified repository file. In a development-to-production scenario, you have an original parent file, a current file that contains the latest development changes, and a modified file that is the deployed copy of the original file. To generate a patch, you open the current file, select the original file, and then create the patch.

The example in the slide shows how to create a patch in a development-to-production scenario. The original repository (`original.rpd`) is the repository initially created during development and then rolled out to production (`modified.rpd`). Typically, before the patch is applied, the original repository and the production repository are identical. The development repository (`current.rpd`) is the repository that contains the changes you want to put in the patch. Assume that this is a repository you have updated after rolling out the original repository (`original.rpd`) to production (`modified.rpd`).

In an Oracle BI Applications repository upgrade scenario, the current file is the latest version of the repository shipped by Oracle, and the original file is the original repository shipped by Oracle. The modified file is the file that contains the customizations you made to the original file.

Applying a Patch



Applying a Patch

To apply a patch, you open the modified file, select the original file, and then apply the patch. The example in the slide shows how to apply a patch in a development-to-production scenario.

The process is the same in an upgrade scenario. Again, in an Oracle BI Applications repository upgrade scenario, the current file is the latest version of the repository shipped by Oracle, and the original file is the original repository shipped by Oracle. The modified file is the file that contains the customizations you made to the original file.

Detailed steps for this process are shown in the following slides.

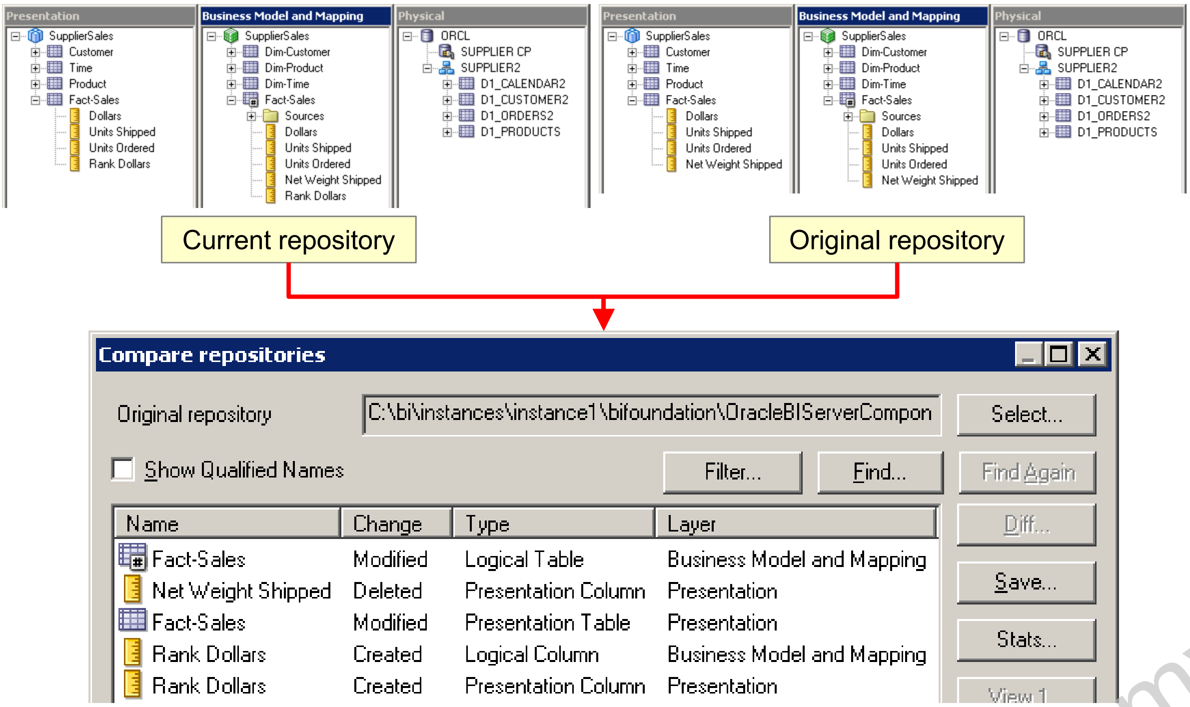
Steps to Perform a Patch Merge

1. Compare current and original repositories.
2. Equalize objects.
3. Create a patch.
4. Apply the patch.
5. Make merge decisions.
6. Verify your work.

Steps to Perform a Patch Merge

This slide shows the high-level steps needed to perform a patch merge. Each step is covered in more detail on subsequent slides.

1. Compare Current and Original Repositories



ORACLE

1. Compare Current and Original Repositories

In the Administration Tool, open the current Oracle BI repository in offline mode. In other words, open the updated repository that contains the changes you want to put in the patch. Assume that this is a repository you have updated after rolling out the original repository (`original.rpd`) to production (`modified.rpd`). Select File > Compare and select the original Oracle BI repository. The "Compare repositories" dialog box appears, compares the two repositories, and marks objects as created, deleted, or modified.

Created: Object was created in the current repository and doesn't exist in the original repository.

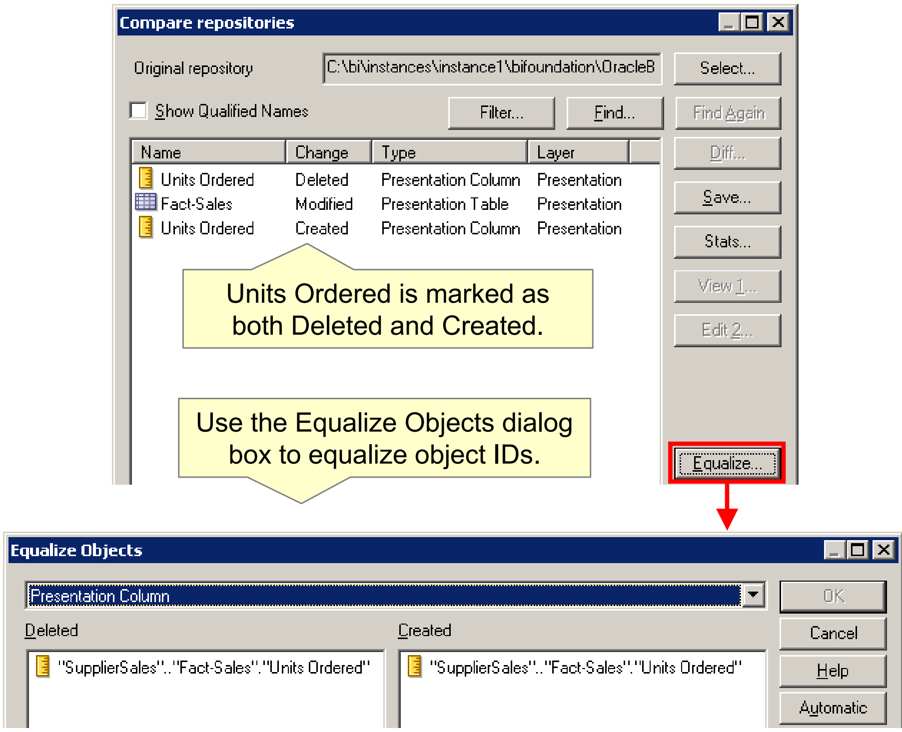
Deleted: Object exists in the original repository but is deleted from the current repository.

Modified: Object exists in the original repository but is modified in the current repository.

In this example, the changes are the following:

- A `Rank Dollars` logical column has been created and added to the `Fact-Sales` logical table in the BMM layer.
- A `Rank Dollars` presentation column has been created and added to the `Fact-Sales` presentation table in the Presentation layer.
- The `Net Weight Shipped` presentation column has been deleted from the `Fact-Sales` presentation table in the Presentation layer.

2. Equalize Objects



ORACLE

2. Equalize Objects

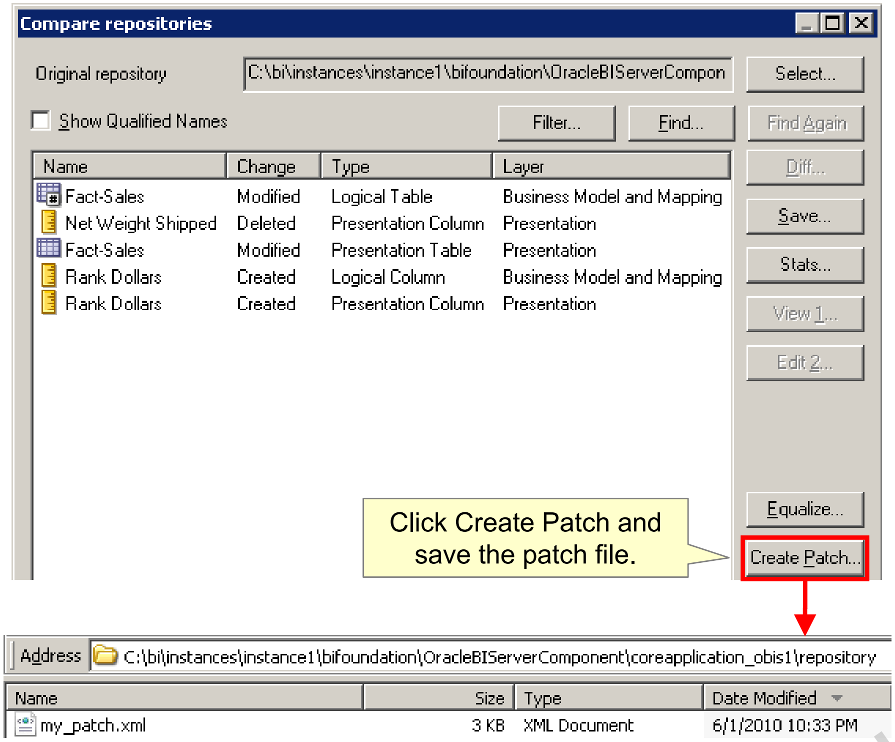
Use the Equalize Objects dialog box in the “Compare repositories” option to compare and equalize repositories. Objects may need to be equalized because the Administration Tool tracks the history of each repository object by using the upgrade ID of the object. Sometimes the upgrade ID can change because of user actions or during merge. When this occurs and a subsequent comparison is done, the Administration Tool treats the new upgrade ID as a new object and the missing original upgrade ID as a deleted object.

In this example, the modified repository is compared to the original repository. Notice that the `Units Ordered` presentation column is marked as both deleted and created. This is because the object was deleted and then re-created in the modified repository. The `Fact-Sales` presentation table is marked as modified because the `Units Ordered` presentation column is in this table.

Notice also that only `Presentation Column` is available in the drop-down list, because only presentation columns have been deleted in this example. If more object types had been deleted, they would be displayed in this list (Logical Table, Logical Column, and so forth).

You use the Equalize Objects dialog box to preview changes that are made if you equalize objects by using the `equalizerpds` utility, which equalizes the objects in both repositories so that the object upgrade IDs are consistent. When objects are equalized, you receive a message that the repositories are identical. Not all steps are shown in this slide.

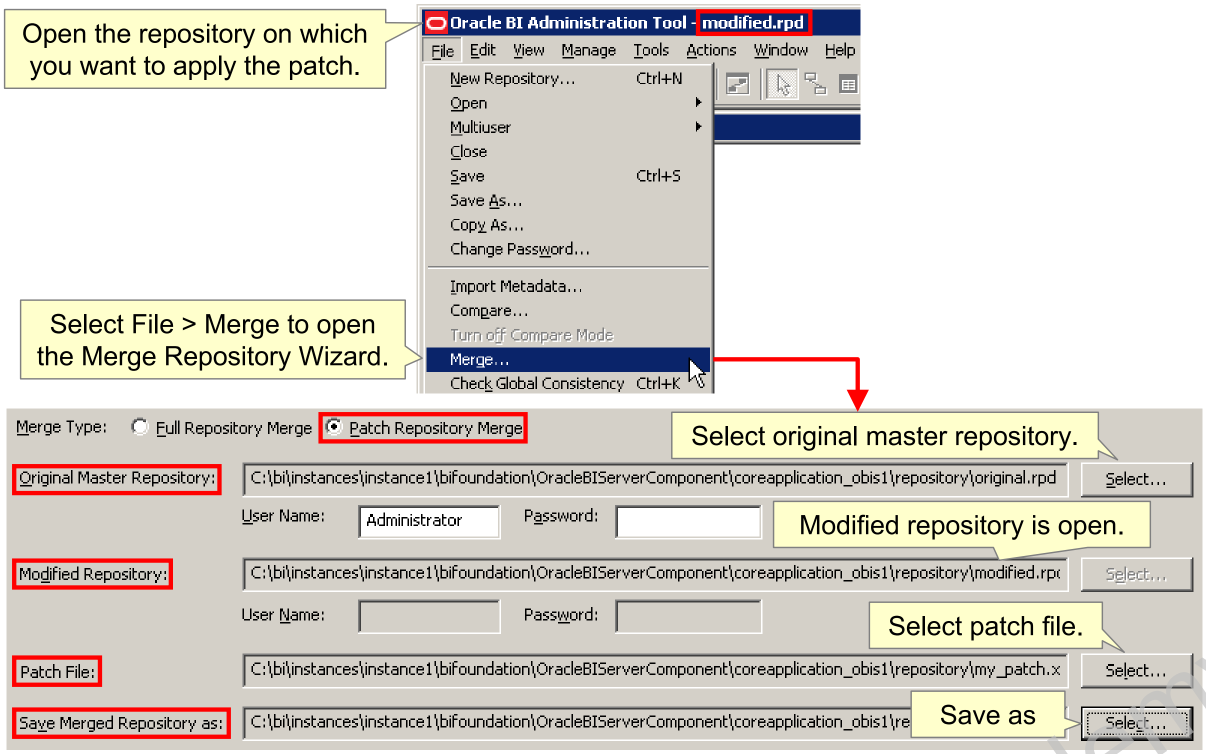
3. Create a Patch



3. Create a Patch

In the “Compare repositories” dialog box, review the changes between the repositories. Then click Create Patch. In the Create Patch dialog box (not shown here), enter a name for the XML patch file and click Save. By default, the file is saved to the repository directory. When the file is saved, the “Compare repositories” dialog box closes automatically.

4. Apply the Patch



4. Apply the Patch

To apply the patch, open the repository on which you want to apply the patch. In this example, the repository is named `modified.rpd`. Select File > Merge to open the Merge Repository Wizard.

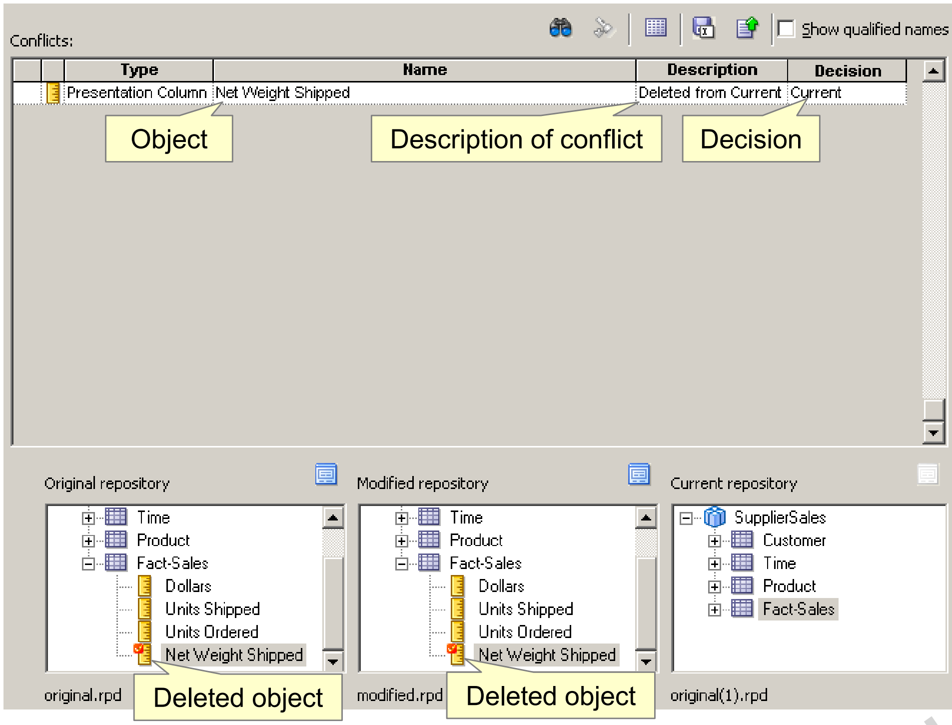
For Merge Type, select Patch Repository Merge. Then select the original master repository. Note that the original repository cannot be the same as the modified repository, which is currently open.

Select the saved patch file.

You can enter an optional name ("Save as") for the merged repository that is created by applying the patch. If you do not enter a name, the system selects one for you.

When you are finished making your selections, click Next (not shown in the screenshot) to open the Merge Repository Wizard.

5. Make Merge Decisions



5. Make Merge Decisions

In the Merge Repository Wizard, you must make decisions for any conflicts between the repositories. Under Conflicts, notice that the description of the conflict for Net Weight Shipped is “Deleted from Current.”

There are two decision choices available in the Decision field:

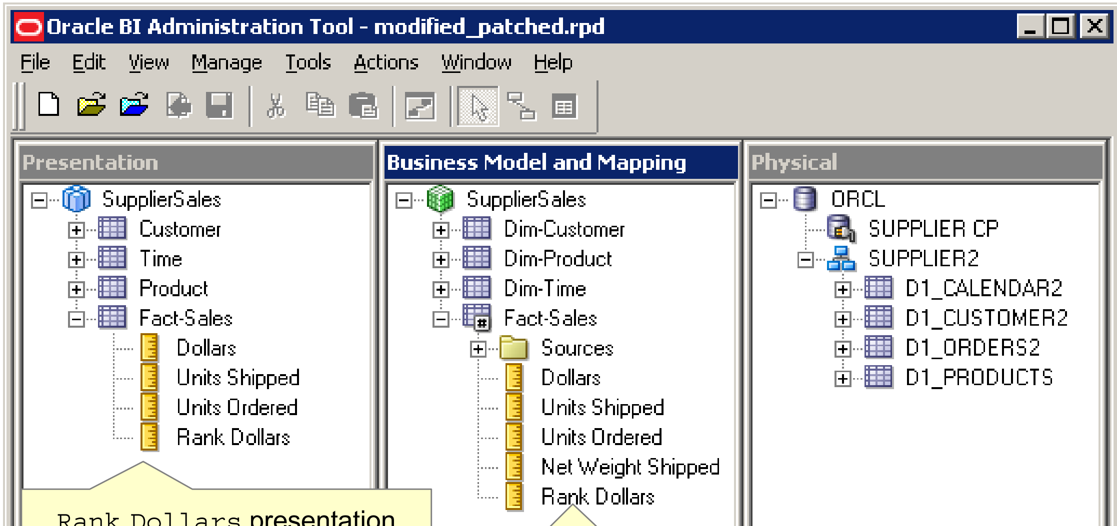
Current: Keeps the repository as it is without adding the object to the merged repository

Modified (A): Adds the object into the merged repository

In this example, you select Current because Net Weight Shipped was deliberately deleted from the current repository and you do not want it added to the merged repository.

Click Finish (not shown here) to close the Merge Repository Wizard and open the merged repository.

6. Verify Your Work



`Rank Dollars` presentation column is added.
`Net Weight Shipped` presentation column is deleted.

`Rank Dollars` logical column is added.

6. Verify Your Work

Verify that the expected changes have been applied. In this example, a `Rank Dollars` logical column and presentation column have been created, and the `Net Weight Shipped` presentation column has been deleted.

Summary

In this lesson, you should have learned how to perform a repository patch merge.

Practice 24-1 Overview: Performing a Patch Merge

This practice covers performing a patch merge in a development-to-production scenario.

Practice 24-1 Overview: Performing a Patch Merge

In a patch merge, you create a patch that contains the differences between the current repository file and the original repository file. Then you apply the patch file to the modified repository file.

In a development-to-production scenario, you have an original parent file, a current file that contains the latest development changes, and a modified file that is the deployed copy of the original file. To generate a patch, you open the current file, select the original file, and then create the patch. To apply the patch, you open the modified file, select the original file, and then apply the patch.

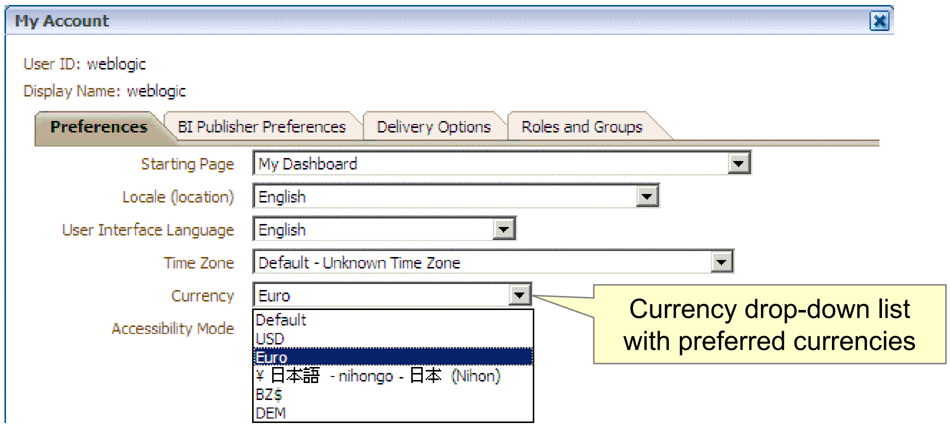
Configuring Logical Columns for Multicurrency Support

Objectives

After completing this lesson, you should be able to configure logical columns for multicurrency support.

Overview

You can configure logical columns so that Oracle BI users can select the currency in which they prefer to view currency columns in analyses and dashboards.



Overview

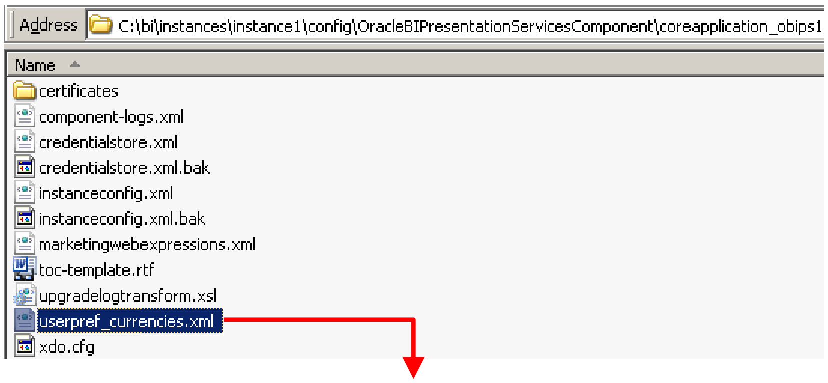
You can set up this feature so that all users see the same static list of currency options, or you can provide a dynamic list of currency options that changes based on a logical SQL statement that you specify. This lesson shows you how to define user-preferred currency options by using a static list of currency options.

Steps to Configure Multicurrency Support

1. Modify the user preferences currency file.
2. Create a `PREFERRED_CURRENCY` session variable.
3. Create logical columns with currency conversions.
4. Edit a logical column to use a conversion factor.
5. Verify your work.

1. Modify the User Preferences Currency File

Modify `userpref_currencies.xml` to enable currency preferences.




```
<UserCurrencyPreferences currencyTagMappingType="static">
  <UserCurrencyPreference sessionVarValue="gc1" displayText="Global Currency 1" currencyTag="int:USD" />
  <UserCurrencyPreference sessionVarValue="gc2" displayText="Global Currency 2" currencyTag="int:euro-l" />
  <UserCurrencyPreference sessionVarValue="gc3" displayText="Global Currency 3" currencyTag="loc:ja-JP" />
  <UserCurrencyPreference sessionVarValue="orgc" displayText="Org Currency" currencyTag="loc:en-BZ" />
  <UserCurrencyPreference sessionVarValue="lcl" displayTag="int:DEM" currencyTag="int:DEM" />
</UserCurrencyPreferences>
```

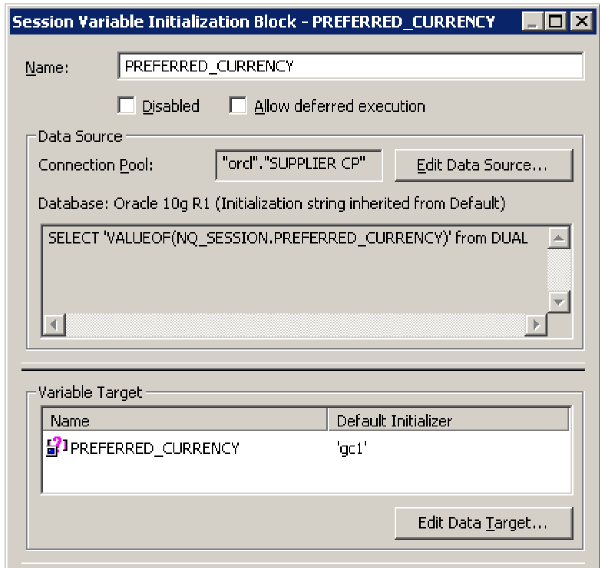

1. Modify the User Preferences Currency File

In the Currency box on the Preferences tab of the My Account dialog box, Oracle Business Intelligence users can select the currency in which they prefer to view currency columns in analyses and dashboards. You use the `userpref_currencies.xml` file to define the currency options that users see in the Currency box.

Navigate to <ORACLE_INSTANCE>\config\OracleBIPresentationServicesComponent\coreapplication_obips1 and open the `userpref_currencies.xml` file for editing. Locate the `UserCurrencyPreferences` element and remove the comment markers `<!--` and `-->` to enable currency preferences.

2. Create a `PREFERRED_CURRENCY` Session Variable

Create a session variable named `PREFERRED_CURRENCY`, along with an initialization block to use the variable.



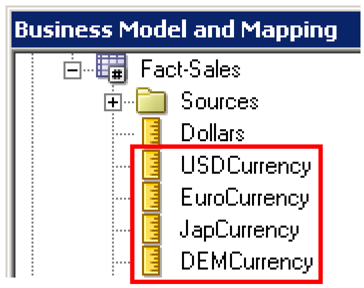
2. Create a `PREFERRED_CURRENCY` Session Variable

Make sure to select "Enable any user" to set the value when you create the session variable (not shown in the slide).

Note: When you use session variables in an expression for Oracle BI Presentation Services, you must preface their names with `NQ_SESSION`.

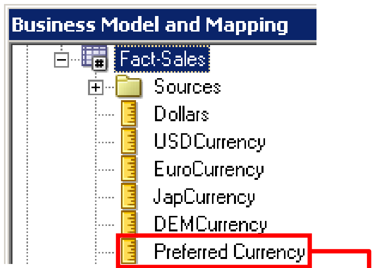
3. Create Logical Columns with Currency Conversions

Typically currency conversion is calculated as part of an extraction, transformation, and loading (ETL) process.



4. Edit a Logical Column to Use a Conversion Factor

Use the `PREFERRED_CURRENCY` session variable to edit any logical columns that display currency values to use the appropriate conversion factor.




```
IndexCol( CASE  VALUEOF(NQ_SESSION."PREFERRED_CURRENCY")
WHEN 'gc1' THEN 0
WHEN 'gc2' THEN 1
WHEN 'gc3' THEN 2
WHEN 'lc1' THEN 3
ELSE 4 END,
"SupplierSales"."Fact-Sales"."USDCurrency",
"SupplierSales"."Fact-Sales"."EuroCurrency",
"SupplierSales"."Fact-Sales"."JapCurrency",
"SupplierSales"."Fact-Sales"."DEMCurrency",
"SupplierSales"."Fact-Sales"."USDCurrency")
```

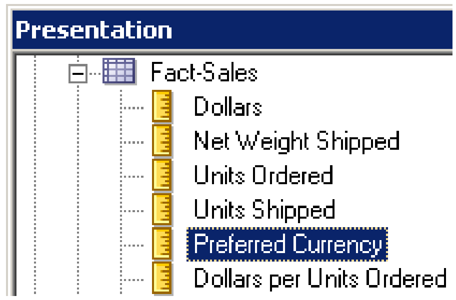

4. Edit a Logical Column to Use a Conversion Factor

Double-click the appropriate logical column in the BMM, click the Column Source tab, and create a derived expression that uses the `PREFERRED_CURRENCY` variable.

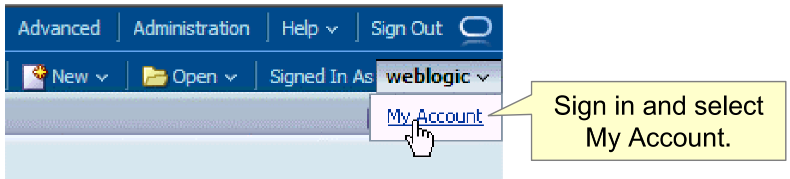
In this example, the `Preferred Currency` logical column expression uses the value of the `NQ_SESSION.PREFERRED_CURRENCY` variable to switch between the different currency columns. Note that the currency columns are expected to have the appropriate converted values.

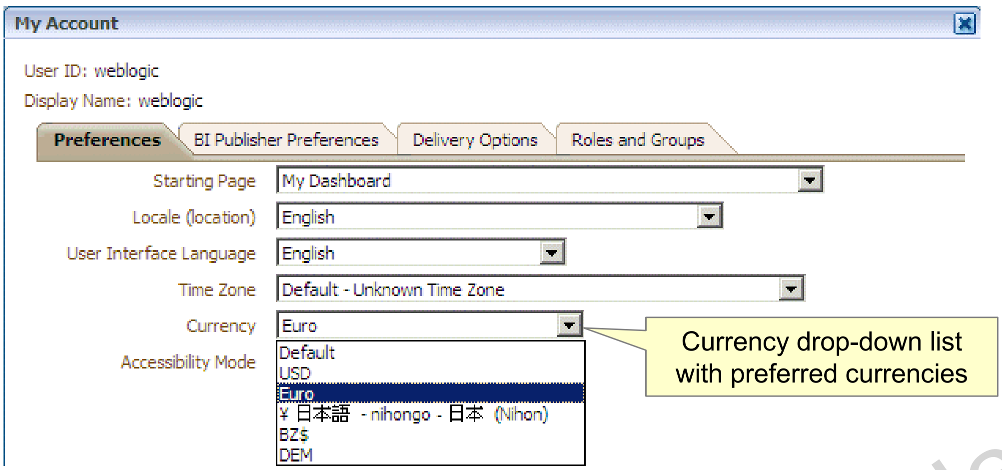
4. Add Logical Column to the Presentation Layer

Add the logical column with the preferred currency expression to the Presentation layer.



5. Verify the My Account Page



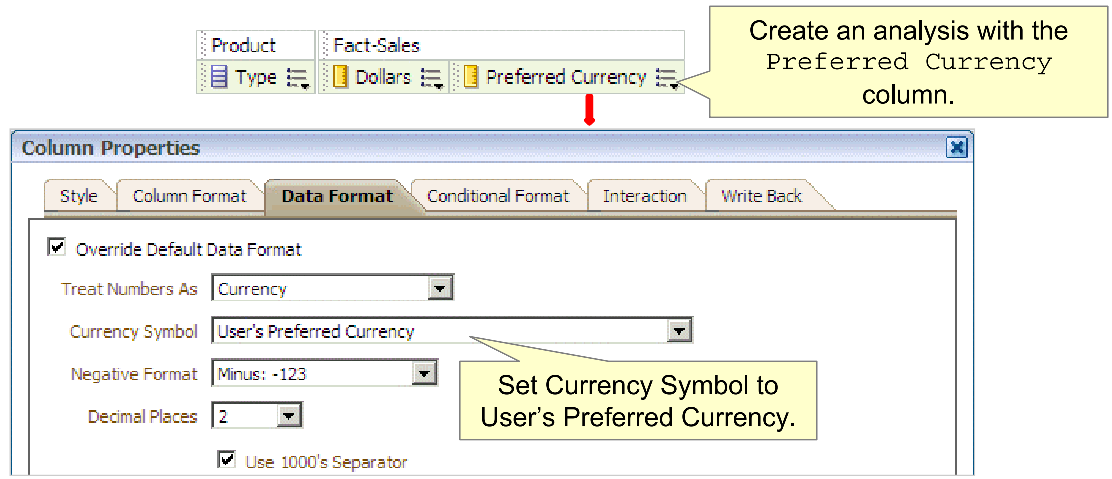


5. Verify the My Account Page

Sign in to Oracle BI and click My Account. Confirm that you can see a drop-down list for Currency with the preferred currencies listed.

In this example, the preferred currency is set to `Euro`.

5. Verify Analysis Results



Type	Dollars	Preferred Currency
Baking	$4,925,521	€ 3,694,140.60
Beef	$4,916,016	€ 3,687,012.15
Beverage	$4,398,107	€ 3,298,580.61
Bread	$1,578,743	€ 1,184,057.59
Cereal	$1,309,071	€ 981,803.56
Cheese	$7,140,616	€ 5,355,462.07
Condiments	$9,105,121	€ 6,828,841.04

Result displays the user's preferred currency.

5. Verify Analysis Results

Create a new analysis with the `Preferred Currency` column.

Select Column Properties > Data Format, override the default data format, and set Currency Symbol to User's Preferred Currency.

When you run the analysis, the `Preferred Currency` column displays the preferred currency that you selected on the My Account page (in this example, the results are displayed in euros).

Summary

In this lesson, you should have learned how to configure logical columns for multicurrency support.

Practice 25-1 Overview: Defining User-Preferred Currency Options

This practice covers defining user-preferred currency options by using a static mapping.

Practice 25-1 Overview: Defining User-Preferred Currency Options

Oracle Business Intelligence users can select the currency in which they prefer to view currency columns in analyses and dashboards in the Currency box on the Preferences tab of the My Account dialog box. You define the currency options that are to be displayed in the Currency box in the `userpref_currencies.xml` file.

26

Using Administration Tool Utilities

Objectives

After completing this lesson, you should be able to:

- Describe the various wizards and utilities in the Oracle BI Administration Tool
- Use the Administration Tool wizards and utilities to manage, maintain, and enhance repositories

Wizards and Utilities

This lesson describes the following utilities and wizards:

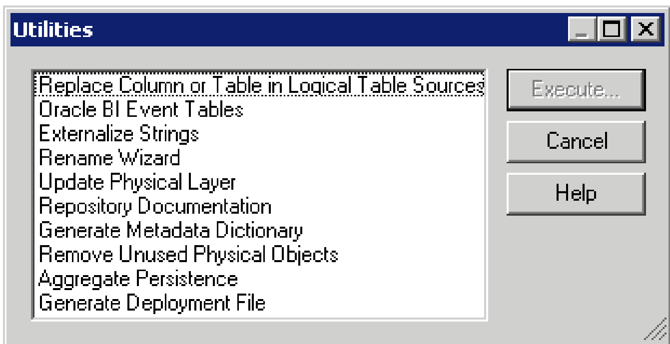
- Session Manager
- Query Repository
- Replace Columns and Tables
- Repository Documentation
- Generate Metadata Dictionary
- Oracle BI Event Tables
- Update Physical Layer
- Remove Unused Physical Objects

Wizards and Utilities

In the process of building the SupplierSales and Inventory business models in this course, you had a chance to interact with a number of features of the Administration Tool. In this lesson, you are introduced to additional Administration Tool utilities and wizards that can aid in the development, maintenance, and administration of repositories.

Accessing Wizards and Utilities

- Select Manage > Sessions to access the Sessions Manager.
- Select Tools > Query Repository to access the Query Repository utility.
- Select Tools > Utilities to access remaining utilities.

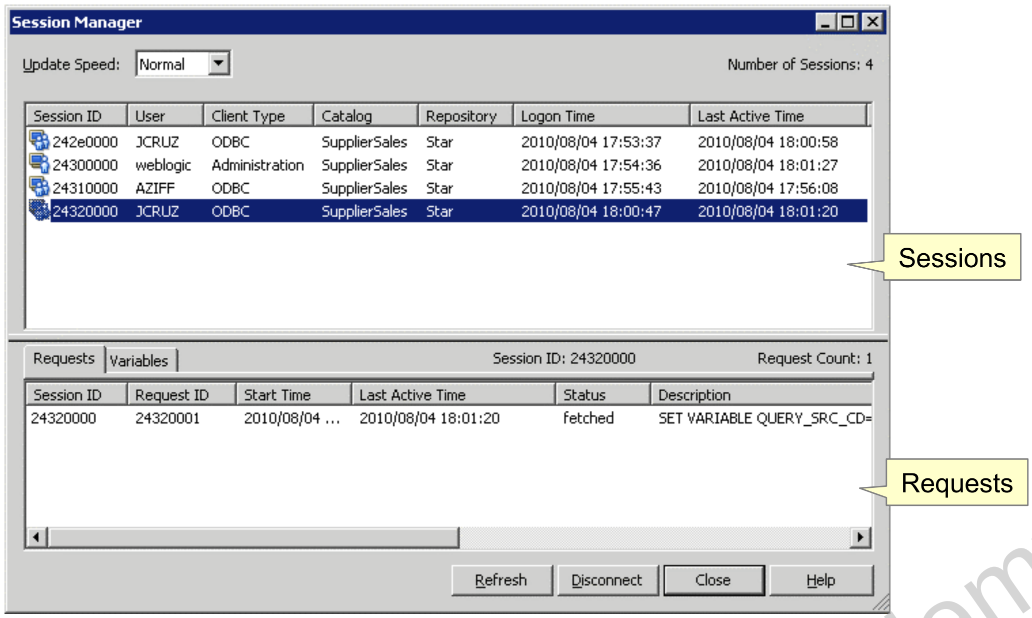


Accessing Wizards and Utilities

Use the Manage and Tools menus to access Administration Tool wizards and utilities. You can also right-click objects in the repository to access some utilities and wizards, such as the Query Repository utility and the Calculation Wizard.

Managing Sessions

Use the Session Manager to monitor session activity in online mode.



Managing Sessions

The Session Manager is used in online mode to monitor session activity. The Session Manager shows all users logged in to the session, all current query requests for each user, and variables and their values for a selected session. Additionally, the Oracle BI Server administrator can disconnect any user and kill any query request with the Session Manager. How often the Session Manager data refreshes depends on the amount of activity on the system. To refresh the display at any time, click Refresh.

Using the Session Manager

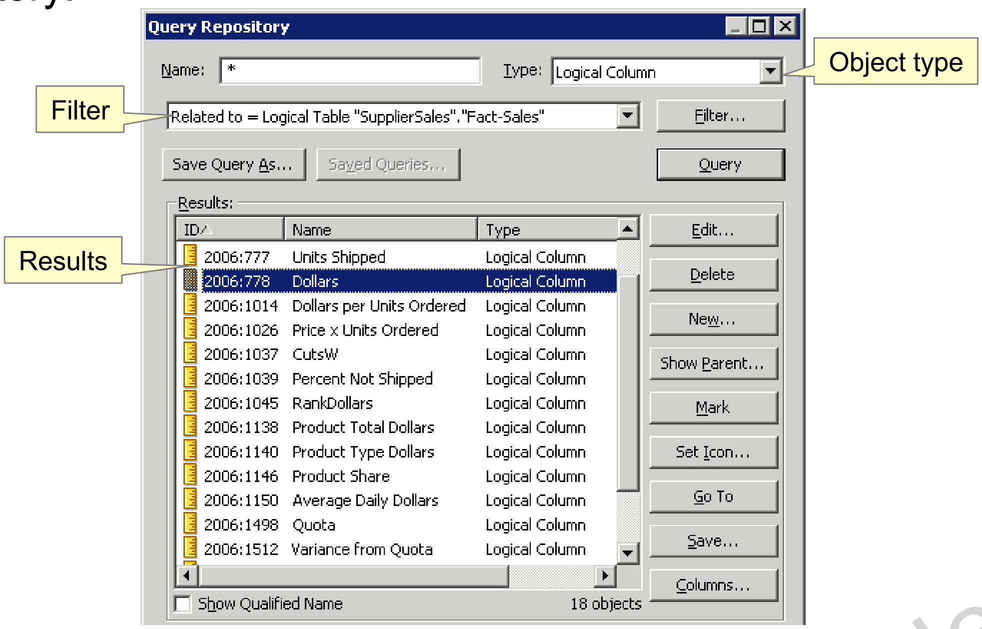
The Session Manager contains upper and lower windows.

The top window, the Session window, shows users currently logged in to Oracle BI Server. To control the update speed, select Normal, High, or Low from the Update Speed drop-down list. Select Pause to keep the display from being refreshed.

The bottom window contains two tabs. The Request tab shows active query requests for the user selected in the Session window. The Variables tab shows variables and their values for a selected session. You can click the column headers to sort the data.

Querying Repository Metadata

Use the Query Repository tool to query for objects in the repository.



Querying Repository Metadata

You can query for objects in the repository by using the Query Repository tool. If you query using the All Types option, you see a listing of the exposed object types in the repository. The list does not contain objects such as aggregate rules, logical source folders, privilege packages, and other objects that are considered to be internal objects.

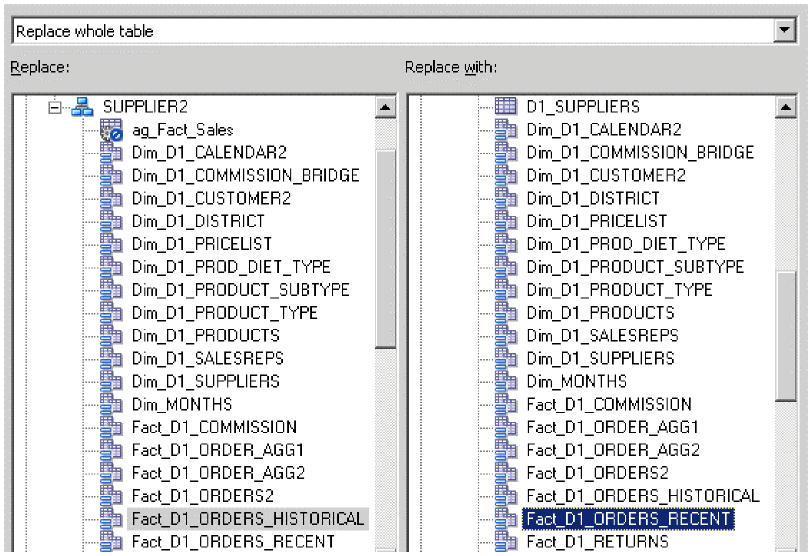
You can use repository queries to help manage the repository metadata in the following ways:

- Examine and update the internal structure of the repository. For example, you can query a repository for objects in the repository based on name, type (such as Catalog, Complex Join, Key, and LDAP Server), or a combination of name and type. You can then edit or delete objects that appear in the Results list. You can also create new objects and view parent hierarchies.
- Query a repository and view reports that show such items as all tables mapped to a logical source, all references to a particular physical column, content filters for logical sources, initialization blocks, and security and user permissions. For example, you might want to run a report prior to making any physical changes in a database that might affect the repository. You can save the report to a file in comma-separated value (CSV) format or tab-delimited format.

When you save results, the encoding options are ANSI, Unicode, and UTF-8.

Replacing Columns or Tables

Use the “Replace Column or Table” utility to automate the process of replacing physical columns and tables in logical table sources.



Replacing Columns or Tables

The “Replace Column or Table” utility automates the process of replacing physical columns and tables in logical table sources by enabling the Oracle BI Server administrator to select the sources from those displayed. The wizard prompts the administrator to replace columns as well as tables.

Documenting a Repository

Use the Repository Documentation utility to document mappings from presentation columns to the corresponding logical and physical columns.

Save in comma-separated, tab-delimited, or XML formats.

Subject Area	Presentation Table	Presentation Column	Business Model	Derived logical table	Derived logical column
SupplierSales	Fact-Sales	Dollars per Units Ordered	SupplierSales	Fact-Sales	Dollars per Units Ordered
SupplierSales	Fact-Sales	Dollars per Units Ordered	SupplierSales	Fact-Sales	Dollars per Units Ordered
SupplierSales	Fact-Sales	Price x Units Ordered	SupplierSales		
SupplierSales	Fact-Sales	Price x Units Ordered	SupplierSales		
SupplierSales	Fact-Sales	CutsW	SupplierSales	Fact-Sales	CutsW
SupplierSales	Fact-Sales	CutsW	SupplierSales	Fact-Sales	CutsW
SupplierSales	Fact-Sales	Percent Not Shipped	SupplierSales	Fact-Sales	Percent Not Shipped
SupplierSales	Fact-Sales	Percent Not Shipped	SupplierSales	Fact-Sales	Percent Not Shipped
SupplierSales	Fact-Sales	RankDollars	SupplierSales	Fact-Sales	RankDollars
SupplierSales	Fact-Sales	Product Total Dollars	SupplierSales	Fact-Sales	Product Total Dollars
SupplierSales	Fact-Sales	Product Type Dollars	SupplierSales	Fact-Sales	Product Type Dollars
SupplierSales	Fact-Sales	Product Share	SupplierSales	Fact-Sales	Product Share
SupplierSales	Fact-Sales	Product Share	SupplierSales	Fact-Sales	Product Total Dollars
SupplierSales	Fact-Sales	Product Share	SupplierSales	Fact-Sales	Product Share

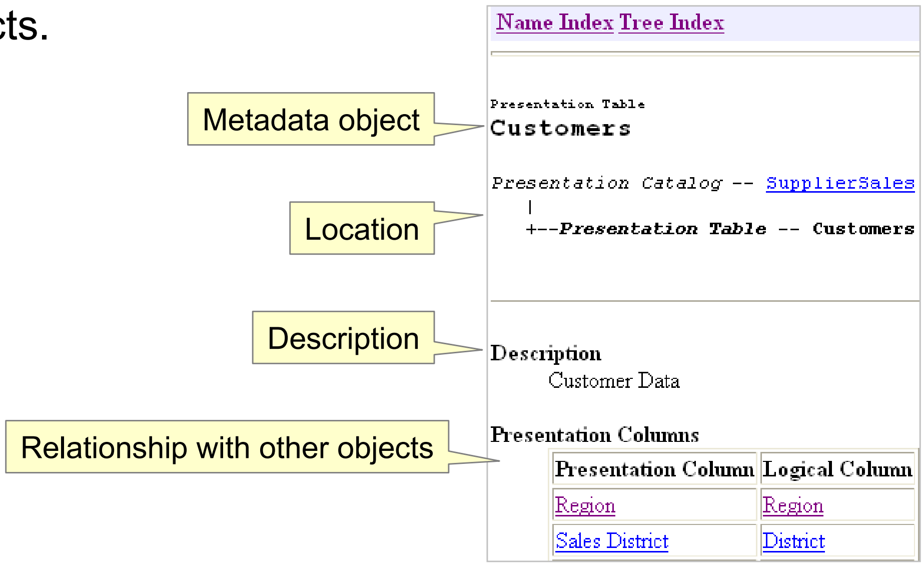
ORACLE

Documenting a Repository

The documentation also includes conditional expressions associated with the columns. If a presentation column derives from several physical columns, there is one row in the document for each physical column.

Generating a Metadata Dictionary

Use the Generate Metadata Dictionary utility to create a set of static XML documents that describe each metadata object, including its properties and its relationships with other metadata objects.

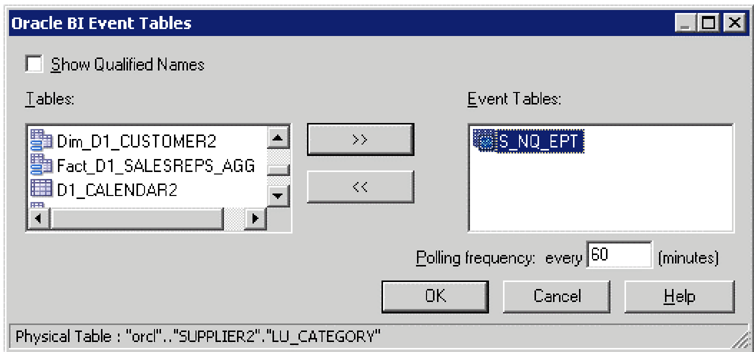


Generating a Metadata Dictionary

This utility provides a user-friendly interface for navigating a repository to help users better understand metadata-object relationships.

Creating an Event Table

Use the Oracle BI Event Tables utility to identify a table as an Oracle BI event polling table.

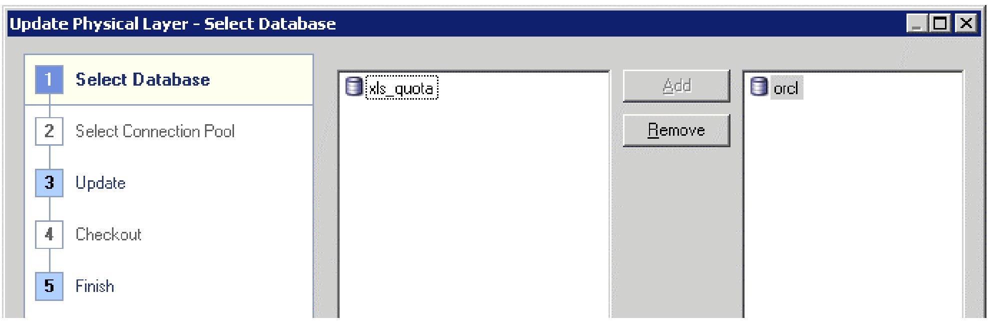


Creating an Event Table

An event polling table is a way to notify Oracle BI Server that one or more physical tables have been updated. Each row that is added to an event table describes a single update event. The cache system reads rows from—or polls—the event table, extracts the physical table information from the rows, and purges cache entries that reference those physical tables. The SQL for creating an event table can be found in the `Oracle BI\server\Schema` folder.

Updating the Physical Layer

Use the Update Physical Layer Wizard to update database objects in the Physical layer of a repository based on their current definitions in the back-end database.



Updating the Physical Layer

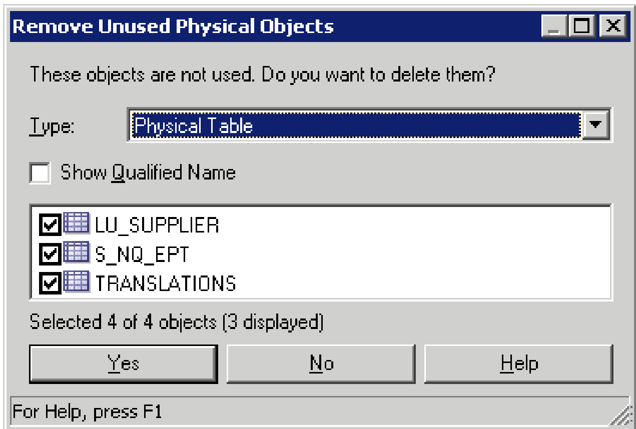
This wizard is not available for repositories that are opened in read-only mode, because they are not available for updating. When the wizard processes the update, the server running the Administration Tool connects to each back-end database. The objects in the Physical layer are compared with those in the back-end database. Explanatory text alerts you to differences between objects defined in the database in the Physical layer and objects defined in the back-end database (such as data type-length mismatches and objects that are no longer found in the back-end database). For example, if an object exists in the database in the Physical layer of the repository but not in the back-end database, the following text is displayed: "Object does not exist in the database."

The wizard does not add columns or tables to the repository if they exist in the back-end database but not in the repository. Additionally, the wizard does not update column key assignments. It checks that there is a column in the repository that matches the column in the database, and then, if the values do not match, the wizard updates the type and length of the column in the repository.

The connection pool settings for each database need to match the connection pool settings used when the objects were last imported into the Physical layer from the back-end database. For example, for Oracle, the connection pool may be set to native Oracle Call Interface (OCI), but an Oracle ODBC source must be used for the update. In this case, you would set the connection pool to the Oracle ODBC setting used for the import.

Removing Unused Physical Objects

Use the Remove Unused Physical Objects utility to remove objects that are no longer needed in a repository.



Removing Unused Physical Objects

Large repositories use more memory on the server and are harder to maintain. Additionally, development activities take longer on a large repository. The Remove Unused Physical Objects utility enables you to remove objects that you no longer need in your repository. You can remove databases, initialization blocks, physical catalogs, and variables.

Summary

In this lesson, you should have learned how to:

- Describe the various wizards and utilities contained in the Oracle BI Administration Tool
- Use the Administration Tool wizards and utilities to manage, maintain, and enhance repositories

Oracle BI 11*g* R1: Build Repositories

Quizzes

Lesson 2: Repository Basics

Question 1

Identify the **correct** statement about the Oracle BI Server architecture.

1. Oracle BI Server is a set of query clients that display the results of analyses.
2. Oracle BI Presentation Services stores the metadata used by Oracle BI Server.
3. Oracle BI Server generates dynamic SQL to query data in physical data sources.
4. Oracle BI Repository can be in multiple formats.

Question 1: Solution

Identify the **correct** statement about the Oracle BI Server architecture.

1. Oracle BI Server is a set of query clients that display the results of analyses.
2. Oracle BI Presentation Services stores the metadata used by Oracle BI Server.
3. Oracle BI Server generates dynamic SQL to query data in physical data sources.
4. Oracle BI Repository can be in multiple formats.

Question 2

Identify the **incorrect** statement about the Oracle BI Repository.

1. Stores the metadata used by Oracle BI Server
2. Is accessed and configured using the Oracle BI Administration Tool
3. Imports metadata from databases and other data sources
4. Is set of graphical tools used to view, build, and modify analyses

Question 2: Solution

Identify the **incorrect** statement about the Oracle BI Repository.

1. Stores the metadata used by Oracle BI Server
2. Is accessed and configured using the Oracle BI Administration Tool
3. Imports metadata from databases and other data sources
4. Is a set of graphical tools used to view, build, and modify analyses

Question 3

Identify the **incorrect** statement about the Oracle BI Repository.

1. The Business Model and Mapping layer objects define the interface that users see to query data from data sources.
2. The Physical layer contains objects representing the physical data sources to which Oracle BI Server submits queries.
3. The Presentation layer contains objects that provide a customized view of a business model to users.
4. The Business Model and Mapping layer is where physical schemas are simplified and reorganized to form the basis of the users' view of the data.

Question 3: Solution

Identify the **incorrect** statement about the Oracle BI Repository.

1. The Business Model and Mapping layer objects define the interface that users see to query data from data sources.
2. The Physical layer contains objects representing the physical data sources to which Oracle BI Server submits queries.
3. The Presentation layer contains objects that provide a customized view of a business model to users.
4. The Business Model and Mapping layer is where physical schemas are simplified and reorganized to form the basis of the users’ view of the data.

Question 4

A repository can be opened for editing only in offline mode.

- True
- False

Question 4: Solution

A repository can be opened for editing only in offline mode.

- True
- False. You can open a repository for editing in online or offline mode. You can perform different tasks based on the mode in which you open the repository.

Lesson 3: Building the Physical Layer of a Repository

Question 1

Identify the **incorrect** statement about the Physical layer.

1. May contain multiple data sources
2. Is typically the last layer built in the repository
3. Contains objects representing the physical data sources to which Oracle BI Server submits queries
4. Is populated by importing metadata from databases and other data sources

Question 1: Solution

Identify the **incorrect** statement about the Physical layer.

1. May contain multiple data sources
2. Is typically the last layer built in the repository
3. Contains objects representing the physical data sources to which Oracle BI Server submits queries
4. Is populated by importing metadata from databases and other data sources

Question 2

Identify the **correct** statement about the connection pool.

1. Is the highest-level object in the Physical layer
2. Defines the features that Oracle BI Server uses with a data source
3. Is an optional display folder that contains tables and columns for a physical schema
4. Defines how Oracle BI Server connects to a data source

Question 2: Solution

Identify the **correct** statement about the connection pool.

1. Is the highest-level object in the Physical layer
2. Defines the features that Oracle BI Server uses with a data source
3. Is an optional display folder that contains tables and columns for a physical schema
4. Defines how Oracle BI Server connects to a data source

Question 3

Physical joins must be defined in a data source prior to import into the Physical layer.

- True
- False

Question 3: Solution

Physical joins must be defined in a data source prior to import into the Physical layer.

- True
- False. Physical joins need not be defined in a data source prior to import into the Physical layer. Joins that already exist in the physical data sources are imported automatically into the Physical layer when you select keys and foreign keys in the Import Metadata Wizard. All other physical joins need to be explicitly defined in the Physical layer.

Question 4

Alias tables must be defined for all physical tables in the Physical layer.

- True
- False

Question 4: Solution

Alias tables must be defined for all physical tables in the Physical layer.

- True
- False. Creating alias tables is not a required step. However, it is recommended that you use table aliases frequently in the Physical layer to eliminate extraneous joins and to include best-practice naming conventions for physical table names.

Lesson 4: Building the Business Model and Mapping Layer of a Repository

Question 1

Identify the **correct** statement about business models.

1. There can be only one business model in the Business Model and Mapping layer.
2. Business model objects always map to only one object in the Physical layer.
3. A business model object is the highest-level object in the Business Model and Mapping layer.
4. Business models can be created only by dragging objects from the Physical layer.

Question 1: Solution

Identify the **correct** statement about business models.

1. There can be only one business model in the Business Model and Mapping layer.
2. Business model objects always map to only one object in the Physical layer.
3. A business model object is the highest-level object in the Business Model and Mapping layer.
4. Business models can be created only by dragging objects from the Physical layer.

Question 2

Identify the **incorrect** statement about logical table sources.

1. Logical table sources always map to only one physical source.
2. Logical tables can have multiple logical table sources.
3. Logical table sources define the mappings from a logical table to a physical table.
4. Logical table sources are created automatically when you drag a table from the Physical layer to a business model.

Question 2: Solution

Identify the **incorrect** statement about logical table sources.

1. Logical table sources always map to only one physical source.
2. Logical tables can have multiple logical table sources.
3. Logical table sources define the mappings from a logical table to a physical table.
4. Logical table sources are created automatically when you drag a table from the Physical layer to a business model.

Question 3

For a valid repository, logical primary keys are required for all logical tables in a business model.

- True
- False

Question 3: Solution

For a valid repository, logical primary keys are required for all logical tables in a business model.

- True
- False. Logical primary keys are required only for each logical dimension table, not for logical fact tables.

Question 4

Logical joins are not required in a business model.

- True
- False

Question 4: Solution

Logical joins are not required in a business model.

- True
- False. Logical joins express the cardinality relationships between logical tables and are a requirement for a valid business model. Specifying logical table joins is required so that Oracle BI Server has the necessary metadata to translate logical requests against the business model into SQL queries against the physical data sources.

Lesson 5: Building the Presentation Layer of a Repository

Question 1

Identify the **incorrect** statement about the Presentation layer.

1. Provides a way to present a customized view of a business model to users
2. Exposes only the data that is meaningful to users
3. Organizes the data in a way that aligns with the way users think about the data
4. Uses the same object names as those in the Business Model and Mapping layer

Question 1: Solution

Identify the **incorrect** statement about the Presentation layer.

1. Provides a way to present a customized view of a business model to users
2. Exposes only the data that is meaningful to users
3. Organizes the data in a way that aligns with the way users think about the data
4. Uses the same object names as those in the Business Model and Mapping layer

Question 2

Identify the **correct** statement about subject areas.

1. A single subject area must be populated with content from multiple business models.
2. Multiple subject areas cannot reference the same business model.
3. Subject areas organize and simplify a business model for a set of users.
4. Subject areas must map directly to tables in the Physical layer.

ORACLE

Question 2: Solution

Identify the **correct** statement about subject areas.

1. A single subject area must be populated with content from multiple business models.
2. Multiple subject areas cannot reference the same business model.
3. Subject areas organize and simplify a business model for a set of users.
4. Subject areas must map directly to tables in the Physical layer.

Question 3

Identify the **incorrect** statement about presentation tables.

1. Organize presentation columns into categories that make sense to the users
2. May contain columns from one or more logical tables
3. Can be modified independently of logical tables
4. Can be created only by dragging logical tables from the BMM layer

ORACLE

Question 3: Solution

Identify the **incorrect** statement about presentation tables.

1. Organize presentation columns into categories that make sense to the users
2. May contain columns from one or more logical tables
3. Can be modified independently of logical tables
4. Can be created only by dragging logical tables from the BMM layer

ORACLE

Question 4

A presentation table has a one-to-one mapping to a logical table.

- True
- False

Question 4: Solution

A presentation table has a one-to-one mapping to a logical table.

- True
- False. A presentation table does not necessarily have a one-to-one mapping to a logical table. Presentation tables are used for organizing columns and can be organized and structured differently from logical tables in the BMM layer.

Lesson 6: Testing and Validating a Repository

Question 1

Which of the following is not a requirement for a consistent repository?

1. All logical columns are mapped directly or indirectly to one or more physical columns.
2. All logical dimension tables have a logical key.
3. All logical tables have a logical join relationship to another logical table.
4. Logging must be enabled for at least one user.
5. A subject area exists for the business model.

Question 1: Solution

Which of the following is not a requirement for a consistent repository?

1. All logical columns are mapped directly or indirectly to one or more physical columns.
2. All logical dimension tables have a logical key.
3. All logical tables have a logical join relationship to another logical table.
4. Logging must be enabled for at least one user.
5. A subject area exists for the business model.

Question 2

Best Practices consistency messages must be fixed before publishing a repository.

- True
- False

ORACLE

Question 2: Solution

Best Practices consistency messages must be fixed before publishing a repository.

- True
- False. Best Practice messages provide information about conditions but do not indicate an inconsistency.

Question 3

Identify the **incorrect** statement about publishing a repository.

1. You use Fusion Middleware Control Enterprise Manager to publish a repository.
2. Publishing a repository enables Oracle BI Server to load the repository into memory at startup.
3. Publishing a repository makes the repository available for queries by end users.
4. You need to check consistency after a repository is published.

Question 3: Solution

Identify the **incorrect** statement about publishing a repository.

1. You use Fusion Middleware Control Enterprise Manager to publish a repository.
2. Publishing a repository enables Oracle BI Server to load the repository into memory at startup.
3. Publishing a repository makes the repository available for queries by end users.
4. You need to check consistency after a repository is published.

Question 4

Identify the **incorrect** statement about query logging.

1. Oracle BI Server provides a facility for logging query activity at the individual user level.
2. Logging is intended for quality assurance testing, debugging, and use by Oracle Technical Support.
3. Logging levels are typically set to 6 or above for Oracle BI administrators.
4. Query logging is normally disabled in production mode.

Question 4: Solution

Identify the **incorrect** statement about query logging.

1. Oracle BI Server provides a facility for logging query activity at the individual user level.
2. Logging is intended for quality assurance testing, debugging, and use by Oracle Technical Support.
3. Logging levels are typically set to 6 or above for Oracle BI administrators.
4. Query logging is normally disabled in production mode.

Lesson 7: Managing Logical Table Sources

Question 1

You should add a new logical table source if data is duplicated across physical tables.

- True
- False

Question 1: Solution

You should add a new logical table source if data is duplicated across physical tables.

- True
- False

Question 2

Dragging physical columns to an existing logical table source automatically creates logical columns in a business model.

- True
- False

Question 2: Solution

Dragging physical columns to an existing logical table source automatically creates logical columns in a business model.

- True
- False

Question 3

Dragging physical columns to a logical table source automatically adds physical table mappings to the logical table source.

- True
- False

Question 3: Solution

Dragging physical columns to a logical table source automatically adds physical table mappings to the logical table source.

- True
- False

Question 4

You should add multiple sources to a logical table source if data is not duplicated across tables.

- True
- False

Question 4: Solution

You should add multiple sources to a logical table source if data is not duplicated across tables.

- True
- False

Lesson 8: Adding Calculations to a Fact

Question 1

Which of the following is not a method for creating calculation measures in an Oracle BI business model?

1. Use the Expression Builder to create calculation measures by using existing logical columns as objects in a formula.
2. Use the Expression Builder to create calculation measures by using physical columns as objects in the formula.
3. Use the Calculation Wizard to create calculation measures derived from existing presentation columns.

Question 1: Solution

Which of the following is not a method for creating calculation measures in an Oracle BI business model?

1. Use the Expression Builder to create calculation measures by using existing logical columns as objects in a formula.
2. Use the Expression Builder to create calculation measures by using physical columns as objects in the formula.
3. Use the Calculation Wizard to create calculation measures derived from existing presentation columns.

Question 2

You use physical columns for calculations that require an aggregation rule to be applied *after* the calculation.

- True
- False

Question 2: Solution

You use physical columns for calculations that require an aggregation rule to be applied *after* the calculation.

- True
- False

Question 3

You use logical columns for calculation formulas that require an aggregation rule that is applied *before* the calculation.

- True
- False

ORACLE

Question 3: Solution

You use logical columns for calculation formulas that require an aggregation rule that is applied *before* the calculation.

- True
- False

Question 4

The Calculation Wizard uses physical columns as objects in its calculations. As a result, aggregation rules are applied after the calculation.

- True
- False

Question 4: Solution

The Calculation Wizard uses physical columns as objects in its calculations. As a result, aggregation rules are applied after the calculation.

- True
- False. The Calculation Wizard uses *logical* columns as objects in its calculations, so aggregation rules are applied *before* the calculation.

Lesson 9: Creating Logical Dimensions

Question 1

Identify the **incorrect** statement about logical dimensions.

1. Introduce formal hierarchies into a business model
2. Establish levels for data groupings and calculations
3. Are columns whose values are calculated to a specific level of aggregation
4. Provide paths for drill down

Question 1: Solution

Identify the **incorrect** statement about logical dimensions.

1. Introduce formal hierarchies into a business model
2. Establish levels for data groupings and calculations
3. Are columns whose values are calculated to a specific level of aggregation
4. Provide paths for drill down

Question 2

Level-based hierarchies are those in which members of the same type occur only at a single level.

- True
- False

Question 2: Solution

Level-based hierarchies are those in which members of the same type occur only at a single level.

- True
- False

Question 3

Identify the **incorrect** statement about parent-child logical dimensions.

1. A parent-child logical dimension is a hierarchy where members of the same type occur only at a single level.
2. The most common real-life occurrence of a parent-child hierarchy is an organizational reporting hierarchy.
3. A parent-child hierarchy is typically based on a single logical table (for example, the `Employees` table.
4. For each Oracle BI Server parent-child hierarchy defined on a relational table, you must explicitly define the inter-member relationships in a separate parent-child relationship table.

Question 3: Solution

Identify the **incorrect** statement about parent-child logical dimensions.

1. A parent-child logical dimension is a hierarchy where members of the same type occur only at a single level.
2. The most common real-life occurrence of a parent-child hierarchy is an organizational reporting hierarchy.
3. A parent-child hierarchy is typically based on a single logical table (for example, the `Employees` table).
4. For each Oracle BI Server parent-child hierarchy defined on a relational table, you must explicitly define the inter-member relationships in a separate parent-child relationship table.

ORACLE

Question 4

For logical dimensions that contain multiple logical hierarchies, multiple separate presentation hierarchies are created when you drag a logical dimension hierarchy from the BMM layer to a table in the Presentation layer.

- True
- False

Question 4: Solution

For logical dimensions that contain multiple logical hierarchies, multiple separate presentation hierarchies are created when you drag a logical dimension hierarchy from the BMM layer to a table in the Presentation layer.

- True
- False

Lesson 10: Using Aggregates

Question 1

Identify the **incorrect** statement about aggregate tables.

1. Aggregate tables store pre-computed measures that have been aggregated over a set of dimensional attributes.
2. Using aggregate tables is a popular technique for speeding up response time.
3. An aggregate table should have more rows than the non-aggregate table.

Question 1: Solution

Identify the **incorrect** statement about aggregate tables.

1. Aggregate tables store pre-computed measures that have been aggregated over a set of dimensional attributes.
2. Using aggregate tables is a popular technique for speeding up response time.
3. An aggregate table should have more rows than the non-aggregate table.

Question 2

Oracle BI Server allows queries to use the information stored in aggregate tables automatically, without query authors or query tools having to specify aggregate tables in their queries.

- True
- False

Question 2: Solution

Oracle BI Server allows queries to use the information stored in aggregate tables automatically, without query authors or query tools having to specify aggregate tables in their queries.

- True
- False

Question 3

Identify the key difference between modeling aggregate tables and modeling other source data.

1. You must create a data source connection in the Physical layer.
2. You must add sources to logical tables.
3. You must import physical sources.
4. You must specify aggregation content on the Content tab of a logical table source.

Question 3: Solution

Identify the key difference between modeling aggregate tables and modeling other source data.

1. You must create a data source connection in the Physical layer.
2. You must add sources to logical tables.
3. You must import physical sources.
4. You must specify aggregation content on the Content tab of a logical table source.

Question 4

If a user queries for a level lower than the aggregation content levels specified on the Content tab, Oracle BI Server accesses the detail tables.

- True
- False

Question 4: Solution

If a user queries for a level lower than the aggregation content levels specified on the Content tab, Oracle BI Server accesses the detail tables.

- True
- False

Lesson 11: Using Partitions and Fragments

Question 1

Identify the **incorrect** statement about a partition.

1. Is a database element that contains part of the data for a fact or a dimension
2. Must include quota or sales data
3. Combines with other data fragments as necessary
4. May be fact based, value based, level based, or complex

Question 1: Solution

Identify the **incorrect** statement about a partition.

1. Is a database element that contains part of the data for a fact or a dimension
2. Must include quota or sales data
3. Combines with other data fragments as necessary
4. May be fact based, value based, level based, or complex

Question 2

Data is partitioned by fact when the data is split into separate tables according to the levels of the data.

- True
- False

Question 2: Solution

Data is partitioned by fact when the data is split into separate tables according to the levels of the data.

- True
- False. Data is partitioned by fact when different fact data is stored in different tables.

Question 3

When a logical table source does not contain the entire set of data at a given level, you use the “Fragmentation content” edit box on the Content tab of the Logical Table Source window to specify the portion, or fragment, of the set of data that it contains.

- True
- False

Question 3: Solution

When a logical table source does not contain the entire set of data at a given level, you use the “Fragmentation content” edit box on the Content tab of the Logical Table Source window to specify the portion, or fragment, of the set of data that it contains.

- True
- False

Lesson 12: Using Repository Variables

Question 1

Identify the **incorrect** statement about variables.

1. Contain values in memory that are used by Oracle BI Server during its processing
2. Are created and managed using the Variable Manager feature in the Administration Tool
3. Cannot be used to modify metadata content dynamically to adjust to a changing data environment.
4. Consist of two classes: session and repository

Question 1: Solution

Identify the **incorrect** statement about variables.

1. Contain values in memory that are used by Oracle BI Server during its processing
2. Are created and managed using the Variable Manager feature in the Administration Tool
3. Cannot be used to modify metadata content dynamically to adjust to a changing data environment
4. Consist of two classes: session and repository

Question 2

Identify the **incorrect** statement about initialization blocks.

1. Are used to initialize system and non-system session variables, as well as dynamic repository variables
2. Must always determine the latest dates contained in the source data
3. Specify SQL to be run to populate one or more variables by accessing data sources
4. Are invoked at Oracle BI Server startup and are periodically rerun to refresh values for dynamic variables according to an established schedule

Question 2: Solution

Identify the **incorrect** statement about initialization blocks.

1. Are used to initialize system and non-system session variables, as well as dynamic repository variables
2. Must always determine the latest dates contained in the source data
3. Specify SQL to be run to populate one or more variables by accessing data sources
4. Are invoked at Oracle BI Server startup and are periodically rerun to refresh values for dynamic variables according to an established schedule

Question 3

Dynamic repository variables persist only while a user’s session is active.

- True
- False

Question 3: Solution

Dynamic repository variables persist only while a user's session is active.

- True
- False

Question 4

Repository variables can be used instead of literals or constants in the Expression Builder in the Administration Tool.

- True
- False

Question 4: Solution

Repository variables can be used instead of literals or constants in the Expression Builder in the Administration Tool.

- True
- False

Lesson 13: Modeling Time Series Data

Question 1

The best way to make time comparisons is with SQL statements sent directly to the database.

- True
- False

ORACLE

Question 1: Solution

The best way to make time comparisons is with SQL statements sent directly to the database.

- True
- False. There is no direct way to make time comparisons in SQL.

Question 2

Identify the **correct** statement about Oracle BI time series functions.

1. `AGO` aggregates a measure attribute from the beginning of a specified time period to the currently displayed time.
2. `TODATE` calculates aggregated value as of some time period shifted from the current time.
3. `PERIODROLLING` performs an aggregation across a specified set of query grain periods, rather than within a fixed time series grain.

Question 2: Solution

Identify the **correct** statement about Oracle BI time series functions.

1. `AGO` aggregates a measure attribute from the beginning of a specified time period to the currently displayed time.
2. `TODATE` calculates aggregated value as of some time period shifted from the current time.
3. `PERIODROLLING` performs an aggregation across a specified set of query grain periods, rather than within a fixed time series grain.

Question 3

Compared to modeling an ordinary dimension, the time dimension requires which two additional steps?

1. Creating a measure by using the `AGO` function
2. Selecting the Time option in the Logical Dimension dialog box
3. Identifying the time series grain
4. Designating a chronological key for every level of every dimension hierarchy

Question 3: Solution

Compared to modeling an ordinary dimension, the time dimension requires which two additional steps:

1. Creating a measure by using the `AGO` function
2. Selecting the Time option in the Logical Dimension dialog box
3. Identifying the time series grain
4. Designating a chronological key for every level of every dimension hierarchy

Question 4

Identify the **correct** statement about query grain.

1. It is the grain at which the aggregation or offset is requested for both the `AGO` and the `TODATE` functions.
2. It is the grain of the source.
3. It is the lowest time grain of the request.

Question 4: Solution

Identify the **correct** statement about query grain.

1. It is the grain at which the aggregation or offset is requested for both the `AGO` and the `TODATE` functions.
2. It is the grain of the source.
3. It is the lowest time grain of the request.

Lesson 14: Modeling Many-to-Many Relationships

Question 1

Identify the **incorrect** statement about a bridge table.

1. Resolves many-to-many relationships between dimension tables and fact tables
2. Stores multiple records corresponding to a fact
3. Contains a weight factor column representing the ratio of the many-to-many relationship
4. Stores multiple records corresponding to a dimension

ORACLE

Question 1: Solution

Identify the **incorrect** statement about a bridge table.

1. Resolves many-to-many relationships between dimension tables and fact tables
2. Stores multiple records corresponding to a fact
3. Contains a weight factor column representing the ratio of the many-to-many relationship
4. Stores multiple records corresponding to a dimension

Question 2

When modeling a bridge table, you should map the fact logical table source to the bridge table.

- True
- False

Question 2: Solution

When modeling a bridge table, you should map the fact logical table source to the bridge table.

- True
- False

Lesson 15: Localizing Oracle BI Metadata

Question 1

Which of the following are necessary steps when configuring Oracle BI metadata?

1. Using a translation table to store language-specific names and descriptions for metadata
2. Configuring database data, such as product names
3. Using initialization blocks and variables to select the preferred language of the user and the corresponding names and descriptions
4. Configuring chart labels

Question 1: Solution

Which of the following are necessary steps when configuring Oracle BI metadata?

1. Using a translation table to store language-specific names and descriptions for metadata
2. Configuring database data such as product names
3. Using initialization blocks and variables to select the preferred language of the user and the corresponding names and descriptions
4. Configuring chart labels

Question 2

The `WEBLANGUAGE` session variable is populated when a user selects a language during login.

- True
- False

Question 2: Solution

The `WEBLANGUAGE` session variable is populated when a user selects a language during login.

- True
- False

Question 3

Localization preferences are set in the Administration Tool.

- True
- False

Question 3: Solution

Localization preferences are set in the Administration Tool.

- True
- False. Users can change their localization preferences by selecting Settings > My Account in the Oracle BI user interface.

Question 4

Which of the following is not a step to localize metadata?

1. Externalize metadata objects.
2. Run the Externalize Strings utility.
3. Modify the translation file.
4. Load the translation table.
5. Export the translation table.

Question 4: Solution

Which of the following is not a step to localize metadata?

1. Externalize metadata objects.
2. Run the Externalize Strings utility.
3. Modify the translation file.
4. Load the translation table.
5. Export the translation table.

ORACLE

Lesson 16: Localizing Oracle BI Data

Question 1

Which tables are required for data translation?

1. Language table with a list of available languages
2. Administration table with a list of administrative users
3. Lookup table with translations
4. Product tables in French and English

Question 1: Solution

Which tables are required for data translation?

1. Language table with a list of available languages
2. Administration table with a list of administrative users
3. Lookup table with translations
4. Product tables in French and English

Question 2

What are two common techniques for designing translation lookup tables in a multilingual schema?

1. Lookup table for each product table
2. Lookup table for each physical table
3. Lookup table for each base table
4. Lookup table for each translated field

Question 2: Solution

What are two common techniques for designing translation lookup tables in a multilingual schema?

1. Lookup table for each product table
2. Lookup table for each physical table
3. Lookup table for each base table
4. Lookup table for each translated field

Question 3

Identify the correct statement about a logical lookup column.

1. Defines the necessary metadata for a translation lookup table
2. Populates a variable with a language code based on user log in
3. Includes the lookup function to return a translated value
4. Defines the language when a user logs in to Oracle BI server

Question 3: Solution

Identify the correct statement about a logical lookup column.

1. Defines the necessary metadata for a translation lookup table
2. Populates a variable with a language code based on user log in
3. Includes the lookup function to return a translated value
4. Defines the language when a user logs in to Oracle BI server

Question 4

The lookup table must be added to the Presentation layer.

- True
- False

Question 4: Solution

The lookup table must be added to the Presentation layer.

- True
- False

Lesson 17: Setting an Implicit Fact Column

Question 1

Identify the **incorrect** statement about an implicit fact column.

1. Is a column that is added automatically to dimension-only queries
2. Is a column that is included in the query and shown in the analysis results
3. Provides the ability to set a fact table source for a subject area to ensure the expected results for dimension-only queries
4. Forces Oracle BI Server to select a predetermined fact table source even if it is not the most economical source
5. Specifies a default join path between dimension tables when there are several possible alternatives

Question 1: Solution

Identify the **incorrect** statement about an implicit fact column.

1. Is a column that is added automatically to dimension-only queries
2. Is a column that is included in the query and shown in the analysis results
3. Provides the ability to set a fact table source for a subject area to ensure the expected results for dimension-only queries
4. Forces Oracle BI Server to select a predetermined fact table source even if it is not the most economical source
5. Specifies a default join path between dimension tables when there are several possible alternatives

Question 2

You use the Presentation Column properties box to set an implicit fact column.

- True
- False

Question 2: Solution

You use the Presentation Column properties box to set an implicit fact column.

- True
- False. You use the Subject Area properties box to set an implicit fact column.

Lesson 18: Importing Metadata from Multidimensional Data Sources

Question 1

The primary differences between setting up multidimensional data sources and setting up relational data sources are in the Business Model and Mapping layer.

- True
- False

Question 1: Solution

The primary differences between setting up multidimensional data sources and setting up relational data sources are in the Business Model and Mapping layer.

- True
- False. The primary differences between setting up multidimensional data sources and setting up relational data sources are in the Physical layer.

Question 2

Data from multidimensional sources such as Essbase cannot be used in analyses and Oracle BI interactive dashboards.

- True
- False

Question 2: Solution

Data from multidimensional sources such as Essbase cannot be used in analyses and Oracle BI interactive dashboards.

- True
- False

Question 3

Horizontal federation provides the ability to create reports that can display data from both multidimensional and relational data sources.

- True
- False

Question 3: Solution

Horizontal federation provides the ability to create reports that can display data from both multidimensional and relational data sources.

- True
- False

Lesson 19: Security

Question 1

During installation, which three Oracle BI security controls are preconfigured with initial (default) values to form the default security model?

1. Identity store
2. Policy store
3. Authentication store
4. Credential store

Question 1: Solution

During installation, which three Oracle BI security controls are preconfigured with initial (default) values to form the default security model?

1. Identity store
2. Policy store
3. Authentication store
4. Credential store

Question 2

Identify the **correct** statement about the Oracle WebLogic Server Administration Console.

1. Manages policy store application roles that grant permissions to users, groups, or other application roles
2. Manages permissions for Presentation layer objects and business model objects in the repository
3. Manages users and groups for the embedded LDAP server that serves as the out-of-the-box default identity store

Question 2: Solution

Identify the **correct** statement about the Oracle WebLogic Server Administration Console.

1. Manages policy store application roles that grant permissions to users, groups, or other application roles
2. Manages permissions for Presentation layer objects and business model objects in the repository
3. Manages users and groups for the embedded LDAP server that serves as the out-of-the-box default identity store

Question 3

Identify three default application roles.

1. BIAdministrator
2. BIAuthor
3. BIManager
4. BISupervisor
5. BIConsumer

ORACLE

Question 3: Solution

Identify three default application roles.

1. BIAdministrator
2. BIAuthor
3. BIManager
4. BISupervisor
5. BIConsumer

ORACLE

Question 4

Which tool do you use to set repository object permissions?

1. Fusion Middleware Control Enterprise Manager
2. WebLogic Administration Console
3. Analysis Editor
4. Administration Tool

Question 4: Solution

Which tool do you use to set repository object permissions?

1. Fusion Middleware Control Enterprise Manager
2. WebLogic Administration Console
3. Analysis Editor
4. Administration Tool

Question 5

Which tool do you use to create application roles?

1. Fusion Middleware Control Enterprise Manager
2. WebLogic Administration Console
3. Analysis Editor
4. Administration Tool
5. Interactive Dashboard

Question 5: Solution

Which tool do you use to create application roles?

1. Fusion Middleware Control Enterprise Manager
2. WebLogic Administration Console
3. Analysis Editor
4. Administration Tool
5. Interactive Dashboard

Lesson 20: Cache Management

Question 1

Identify the **incorrect** statement about the Oracle BI Server query cache.

1. Saves the results of queries in cache files
2. Enables Oracle BI Server to satisfy subsequent query requests without having to access back-end databases
3. Improves query performance by fulfilling a query from the database as opposed to searching through the cache
4. Conserves network resources by avoiding a connection to the database server

Question 1: Solution

Identify the **incorrect** statement about the Oracle BI Server query cache.

1. Saves the results of queries in cache files
2. Enables Oracle BI Server to satisfy subsequent query requests without having to access back-end databases
3. Improves query performance by fulfilling a query from the database as opposed to searching through the cache
4. Conserves network resources by avoiding a connection to the database server

Question 2

Which tool is used to configure the cache?

1. WebLogic Administration Console
2. Analysis Editor
3. Presentation Services Catalog
4. Fusion Middleware Control Enterprise Manager

Question 2: Solution

Which tool is used to configure the cache?

1. WebLogic Administration Console
2. Analysis Editor
3. Presentation Services Catalog
4. Fusion Middleware Control Enterprise Manager

Question 3

Which file is used to set cache parameters?

1. `Nqsquery.ini`
2. `Nqsserver.ini`
3. `Nqsconfig.ini`
4. `Nqscache.ini`

Question 3: Solution

Which file is used to set cache parameters?

1. `Nqsquery.ini`
2. `Nqsserver.ini`
3. `Nqsconfig.ini`
4. `Nqscache.ini`

Question 4

Identify the **incorrect** statement about the Cache Manager.

1. View information about the entire query cache or individual cache entries.
2. Show, save, or copy cache SQL.
3. Manually purge the cache entries.
4. It must be opened in offline mode.

Question 4: Solution

Identify the **incorrect** statement about the Cache Manager.

1. View information about the entire query cache or individual cache entries.
2. Show, save, or copy cache SQL.
3. Manually purge the cache entries.
4. It must be opened in offline mode.

Question 5

Identify the **incorrect** statement about seeding the cache.

1. It is the process of pre-populating the cache with queries that are known to generate cache misses.
2. It is the process of pre-populating the cache with queries that are known to generate cache hits.
3. It is performed by running prebuilt queries during off hours or immediately after purging.
4. It helps improve query performance.

Question 5: Solution

Identify the **incorrect** statement about seeding the cache.

1. It is the process of pre-populating the cache with queries that are known to generate cache misses.
2. It is the process of pre-populating the cache with queries that are known to generate cache hits.
3. It is performed by running prebuilt queries during off hours or immediately after purging.
4. It helps improve query performance.

Lesson 21: Managing Usage Tracking

Question 1

Identify the **incorrect** statement about Oracle BI usage tracking.

1. Tracks and stores Oracle BI Server usage at the detailed query level
2. Supports the accumulation of statistics that can be used in database performance optimization
3. Supports the accumulation of statistics that can be used in global appreciation strategies
4. Provides ability to analyze usage results by using end-user reporting tools

Question 1: Solution

Identify the **incorrect** statement about Oracle BI usage tracking.

1. Tracks and stores Oracle BI Server usage at the detailed query level
2. Supports the accumulation of statistics that can be used in database performance optimization
3. Supports the accumulation of statistics that can be used in global appreciation strategies
4. Provides ability to analyze usage results by using end-user reporting tools

Question 2

Oracle recommends using which usage-tracking method?

1. Rotation files
2. Direct insertion
3. Log file
4. Aggregation strategy

Question 2: Solution

Oracle recommends using which usage-tracking method?

1. Rotation files
2. Direct insertion
3. Log file
4. Aggregation strategy

Question 3

Identify the **correct** statement about the `S_NQ_ACCT` table.

1. Uploads the usage-tracking data when queries are run against Oracle BI Server
2. Downloads the usage-tracking data when queries are run against Oracle BI Server
3. Purges the usage-tracking data when queries are run against Oracle BI Server
4. Stores the usage-tracking data when queries are run against Oracle BI Server

Question 3: Solution

Identify the **correct** statement about the `S_NQ_ACCT` table.

1. Uploads the usage-tracking data when queries are run against Oracle BI Server
2. Downloads the usage-tracking data when queries are run against Oracle BI Server
3. Purges the usage-tracking data when queries are run against Oracle BI Server
4. Stores the usage-tracking data when queries are run against Oracle BI Server

Question 4

Which of the following is **not** a usage-tracking parameter or setting in the `NQSConfig.ini` file?

1. `ENABLE`
2. `DIRECT_INSERT`
3. `PHYSICAL_TABLE_NAME`
4. `USAGE_TRACKING_BUFFER`
5. `CONNECTION_POOL`

Question 4: Solution

Which of the following is **not** a usage tracking parameter or setting in the `NQSConfig.ini` file?

1. `ENABLE`
2. `DIRECT_INSERT`
3. `PHYSICAL_TABLE_NAME`
4. `USAGE_TRACKING_BUFFER`
5. `CONNECTION_POOL`

Lesson 22: Setting Up and Using the Multiuser Development Environment

Question 1

Identify the **incorrect** statement about the Oracle BI multiuser development environment.

1. Permits multiple users to work with the repository simultaneously
2. Breaks the repository into manageable pieces known as *concurrent repositories*
3. Changes are saved locally and then merged to the master repository.
4. Adheres to the classic Software Configuration Management (SCM) process

Question 1: Solution

Identify the **incorrect** statement about the Oracle BI multiuser development environment.

1. Permits multiple users to work with the repository simultaneously
2. Breaks the repository into manageable pieces known as *concurrent repositories*
3. Changes are saved locally and then merged to the master repository.
4. Adheres to the classic Software Configuration Management (SCM) process

Question 2

Identify the **incorrect** statement about multiuser development projects.

1. Projects are subsets of repository metadata.
2. Projects consist of Presentation layer subject areas and their associated logical facts, dimensions, application roles, users, variables, and initialization blocks.
3. Projects cannot overlap with other projects.

Question 2: Solution

Identify the **incorrect** statement about multiuser development projects.

1. Projects are subsets of repository metadata.
2. Projects consist of Presentation layer subject areas and their associated logical facts, dimensions, application roles, users, variables, and initialization blocks.
3. Projects cannot overlap with other projects.

ORACLE

Question 3

Before checking out projects, each developer must set up the Administration Tool application to point to the multiuser development directory.

- True
- False

Question 3: Solution

Before checking out projects, each developer must set up the Administration Tool application to point to the multiuser development directory.

- True
- False

Question 4

Which multiuser option commits changes to the master repository?

1. Compare with Original
2. Discard Local Changes
3. Publish to Network
4. Merge Local Changes

Question 4: Solution

Which multiuser option commits changes to the master repository?

1. Compare with Original
2. Discard Local Changes
3. Publish to Network
4. Merge Local Changes

Lesson 23: Configuring Write Back in Analyses

Question 1

Which of the following is **not** a step to configure write back?

1. Enable write back for the connection pool.
2. Enable write back for logical columns.
3. Enable write back for presentation columns.
4. Set write back permissions in the Presentation layer.
5. Enable write back in `instanceconfig.xml`.
6. Create a write back template.

Question 1: Solution

Which of the following is **not** a step to configure write back?

1. Enable write back for the connection pool.
2. Enable write back for logical columns.
3. Enable write back for presentation columns.
4. Set write back permissions in the Presentation layer.
5. Enable write back in `instanceconfig.xml`.
6. Create a write back template.

Question 2

Which permission must be set in the Presentation layer to provide write back for application roles or users?

1. Write Back
2. Default
3. Read/Write
4. Everyone

Question 2: Solution

Which permission must be set in the Presentation layer to provide write back for application roles or users?

1. Write Back
2. Default
3. Read/Write
4. Everyone

Question 3

Identify the **incorrect** statement about the write back template.

1. The write back template is an XML-formatted file that contains SQL statements that are needed to insert and update records in the write back table and columns.
2. To ensure that write back works correctly, you must include (in the WebMessage element of the file) the name of the SQL template that you will specify when you create the write back table.
3. You can create only one write back template for each Oracle BI Server instance.
4. You must specify the connection pool along with the SQL commands to insert and update records.

ORACLE

Question 3: Solution

Identify the **incorrect** statement about the write back template.

1. The write back template is an XML-formatted file that contains SQL statements that are needed to insert and update records in the write back table and columns.
2. To ensure that write back works correctly, you must include (in the WebMessage element of the file) the name of the SQL template that you will specify when you create the write back table.
3. You can create only one write back template for each Oracle BI Server instance.
4. You must specify the connection pool along with the SQL commands to insert and update records.

Question 4

All columns in an analysis are automatically enabled for write back by default.

- True
- False

Question 4: Solution

All columns in an analysis are automatically enabled for write back by default.

- True
- False

Lesson 24: Performing a Patch Merge

Question 1

Identify the **incorrect** statement about creating a patch.

1. In a patch merge, you create a patch that contains the differences between the current repository file and the original repository file.
2. You apply the patch file to the current repository file.
3. In a development-to-production scenario, you have an original parent file, a current file that contains the latest development changes, and a modified file that is the deployed copy of the original file.
4. To generate a patch, you open the current file, select the original file, and then create the patch.

Question 1: Solution

Identify the **incorrect** statement about creating a patch.

1. In a patch merge, you create a patch that contains the differences between the current repository file and the original repository file.
2. You apply the patch file to the current repository file.
3. In a development-to-production scenario, you have an original parent file, a current file that contains the latest development changes, and a modified file that is the deployed copy of the original file.
4. To generate a patch, you open the current file, select the original file, and then create the patch.

Question 2

Which of the following is **not** a step to perform a patch merge?

1. Compare current and original repositories.
2. Equalize objects.
3. Compare current and modified repositories.
4. Create a patch.
5. Apply the patch.
6. Make merge decisions.

Question 2: Solution

Which of the following is **not** a step to perform a patch merge?

1. Compare current and original repositories.
2. Equalize objects.
3. Compare current and modified repositories.
4. Create a patch.
5. Apply the patch.
6. Make merge decisions.

Question 3

You create a patch file in the Merge Repositories dialog box.

- True
- False

Question 3: Solution

You create a patch file in the Merge Repositories dialog box.

- True
- False. You create a patch file by clicking the Create Patch button in the “Compare repositories” dialog box.

Question 4

You apply a patch by using the Merge Repository Wizard.

- True
- False

Question 4: Solution

You apply a patch by using the Merge Repository Wizard.

- True
- False

Lesson 25: Configuring Logical Columns for Multicurrency Support

Question 1

You can configure presentation columns so that Oracle BI users can select the currency in which they prefer to view currency columns in analyses and dashboards.

- True
- False

Question 1: Solution

You can configure presentation columns so that Oracle BI users can select the currency in which they prefer to view currency columns in analyses and dashboards.

- True
- False. You can configure *logical* columns so that Oracle BI users can select the currency in which they prefer to view currency columns in analyses and dashboards.

Question 2

Which of the following is **not** a step in configuring multicurrency support?

- Modify the user preferences currency file.
- Create a `PREFERRED_CURRENCY` session variable.
- Create presentation columns with currency conversions.
- Edit a logical column to use a conversion factor.

Question 2: Solution

Which of the following is **not** a step in configuring multicurrency support?

- Modify the user preferences currency file.
- Create a `PREFERRED_CURRENCY` session variable.
- Create presentation columns with currency conversions.
- Edit a logical column to use a conversion factor.

Question 3

If multicurrency conversion is configured correctly, a Currency drop-down list is visible in My Account.

- True
- False

Question 3: Solution

If multicurrency conversion is configured correctly, a Currency drop-down list is visible in My Account.

- True
- False

Question 4

To create an analysis with the user's preferred currency, set Currency Symbol to User's Preferred Currency on the Data Format tab in Column Properties.

- True
- False

Question 4: Solution

To create an analysis with the user's preferred currency, set Currency Symbol to User's Preferred Currency on the Data Format tab in Column Properties.

- True
- False

Lesson 26: Using Administration Tool Utilities

Question 1

Which task **cannot** be performed using the Query Repository utility?

1. Examine and update the internal structure of the repository.
2. Query for objects in the repository based on object type .
3. Edit or delete objects.
4. Monitor repository session activity.

Question 1: Solution

Which task **cannot** be performed using the Query Repository utility?

1. Examine and update the internal structure of the repository.
2. Query for objects in the repository based on object type .
3. Edit or delete objects.
4. Monitor repository session activity.

Question 2

Why do you use the Repository Documentation utility?

1. To automate the process of replacing physical columns or tables in logical table sources
2. To view reports that show such items as all tables mapped to a logical source, all references to a particular physical column, content filters for logical sources, initialization blocks, and security and user permissions
3. To document mappings from presentation columns to the corresponding logical and physical columns
4. To show variables and their values for a selected session

Question 2: Solution

Why do you use the Repository Documentation utility?

1. To automate the process of replacing physical columns or tables in logical table sources
2. To view reports that show such items as all tables mapped to a logical source, all references to a particular physical column, content filters for logical sources, initialization blocks, and security and user permissions
3. To document mappings from presentation columns to the corresponding logical and physical columns
4. To show variables and their values for a selected session

Question 3

Why do you use the Generate Metadata Dictionary utility?

1. To document mappings from presentation columns to the corresponding logical and physical columns
2. To automate the process of replacing physical columns or tables in logical table sources
3. To create a set of static XML documents that describe each metadata object, including its properties and its relationships with other metadata objects
4. To identify a table as an Oracle BI event polling table

Question 3: Solution

Why do you use the Generate Metadata Dictionary utility?

1. To document mappings from presentation columns to the corresponding logical and physical columns
2. To automate the process of replacing physical columns or tables in logical table sources
3. To create a set of static XML documents that describe each metadata object, including its properties and its relationships with other metadata objects
4. To identify a table as an Oracle BI event polling table

Question 4

The Update Physical Layer Wizard enables you to update database objects in the Physical layer of a repository based on their current definitions in the back-end database.

- True
- False

Question 4: Solution

The Update Physical Layer Wizard enables you to update database objects in the Physical layer of a repository based on their current definitions in the back-end database.

- True
- False

ORACLE

A

Optimizing Query Performance

Objective

After completing this appendix, you should be able to identify techniques to optimize Oracle BI query performance.

Business Challenge

Poor query performance of BI tools is the biggest obstacle to end-user adoption and satisfaction:

- Business users expect near-instantaneous query responses.
- Users waiting for results for more than 5–10 seconds seek other alternatives.
- Users are more tolerant when results are valuable or rare.
- Most users do not overtly complain and silently defect.
- Users may defect even if there is no other credible source and may rely on soft, fallible data, instinct, or best guess.
- Exponential growth of data adds to the challenge of maintaining performance.

Business Challenge

The biggest obstacle to end-user adoption of any business intelligence (BI) tool is the inability to ensure consistently strong query performance. Business users have high expectations of near-instantaneous query responses and low tolerance when their expectations are not met. Additionally, users are not very sympathetic to the technical challenges inherent in servicing dynamic, ad hoc BI environments.

Users are generally more tolerant of performance with particularly compelling or exclusive content than with the ordinary. Frustrated users generally do not complain, but they choose not to return over time. This "quiet defection" is particularly apt in BI tool deployments, in which business users tend to be highly creative with multiple back-door channels to information.
BI users move quickly and silently to softer, more fallible data when their needs are not being met. This is especially true when query performance expectations are not being met.

Compounding the challenge of maintaining performance is the exponential growth of stored data. Some analysts estimate that application data volume will increase by 125% per year. While it is necessary to make this data accessible, the majority of data is not actively used.
In fact, as the historical data itself ages, the need for elaborate details diminishes increasingly.

Business Solution

- Establish realistic performance expectations.
  - Four-second to five-second refresh rates are a reasonable performance cap goal for most BI content.
- Monitor end-user query habits.
  - Requests returning large numbers of records indicate that users are running “extracts” and building offline information silos.
- Be proactive with performance.
  - Include an overtly specified, comprehensive performance optimization stage in every implementation project.

ORACLE

Business Solution

One key to managing end-user query performance is establishing realistic expectations. Business users will always demand instantaneous query performance. However, the actual difference between sub-second and sub-five-second query response times is often negligible to the majority of end users. Conversely, the perceived difference between five-second and ten-second response times can be profound.

BI tools are most effective when supporting “speed of thought” analysis, meaning that the performance latency results for one analysis should not negatively affect subsequent analysis accessed via drilling. Business users are generally accustomed to the latency built into refreshing browser pages while surfing the Web. Faster is almost always better, but four-second to five-second refresh rates are a reasonable performance cap goal for most BI content.

Another important step is to monitor end-user query habits. A warning sign of defection from end-user adoption is an increased use of requests returning large numbers of records. This indicates that users are running “extracts,” building and consolidating offline information silos (spreadsheet applications, departmental data marts, and so on) outside the approved, consistent, and centralized information repository.

Business Solution (continued)

It is also best to be proactive when addressing performance needs. Perhaps the most effective means of proactively managing query performance is to mandate and budget tasks solely targeted at query-performance optimization in all release cycles. Many integrators claim their methodologies seamlessly thread performance tasks throughout requirements, design, and build stages. However, performance issues are too often considered at the end of the process or as an afterthought. In these cases, query performance issues become apparent only after end users have experienced problems. This can cause project slippage and negative first impressions that are difficult to overcome. Project plans without explicitly called-out performance optimization tasks should be treated with a great deal of skepticism.

Oracle BI Features That Optimize Performance

Embedded Oracle BI features and functions optimize end-user query performance:

- Intelligent query generation
- Aggregate navigation
- Data source function shipping
- Multipass SQL generation
- Platform-specific SQL/MDX optimization
- Clustering
- Query caching

Oracle BI Features That Optimize Performance

The Oracle BI Enterprise Edition suite of tools is built with attention given to optimizing end-user query performance. Many performance optimization features and functions exist in the platform, including (but not limited to) those listed in the slide.

If Oracle BI EE metadata is configured correctly, end-user query issues are generally bound by the technical limitations of the underlying data sources. Therefore, the most effective means of optimizing end-user query performance is to deploy a comprehensive aggregate summarization process. This performance optimization technique and others are discussed in the following slides.

Optimizing Query Performance

- Using aggregates
- Using aggregate navigation
- Constraining results using a `WHERE` clause
- Caching
- Limiting the number of initialization blocks
- Limiting select table types
- Modeling dimension hierarchies correctly
- Turning off logging
- Setting query limits
- Pushing calculations to the database
- Exposing materialized views in the Physical layer
- Using database hints

Optimizing Query Performance: Repository

There are many query performance optimization steps that can be taken during and after an Oracle BI implementation. Most of these steps have already been covered in this course. The purpose of this appendix is to revisit and explain these topics in the context of query performance optimization. Each topic is discussed in this context in subsequent slides.

Note: The topics discussed in this appendix concern only those steps that can be taken in Oracle BI EE. They do not address issues related to hardware memory; back-end databases; extraction, transformation, and loading (ETL) software; network bandwidth; or the impact of various computing architectures, topologies, memory sizes, CPUs, and so on.

Using Aggregates

Improves performance by enabling Oracle BI Server to generate queries against smaller, summarized tables

Sales (fact) aggregated to Sales Rep, Product Type, and Month levels

SALESREP	TYPEKEY	PERKEY	UNITSHPD	UNITORDD	DOLLARS	NETWGHTSHPD
ALAN ZIFF	118	200801	258	264	3297.1	4177.48
ALAN ZIFF	118	200802	206	206	2743.48	4164.42
ALAN ZIFF	118	200803	276	284	3376.37	5797.85
ALAN ZIFF	118	200804	293	302	3884.35	5408.16

Customer (dimension) aggregated to Sales Rep level

SALESREP	DISTRICT	REGION
ALAN ZIFF	Northwest	West
ANDREW TAYLOR	UpperMidWest	Central
ANN JOHNSON	Yankee	East
ANNE WILLIAMS	Florida	East

Product (dimension) aggregated to Type level

TYPECODE	ITEMTYPE
100	Baking
101	Beef
102	Beverage
103	Bread

Time (dimension) aggregated to Month level

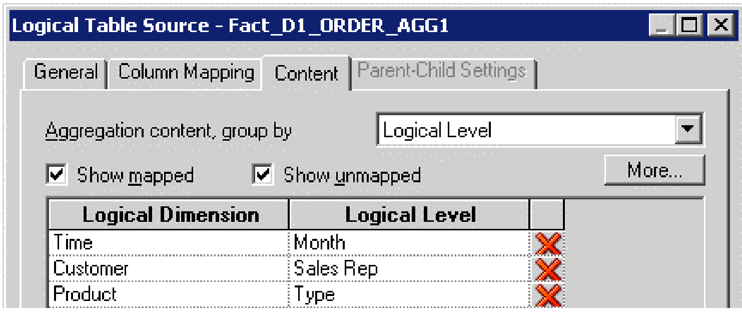
MONTHCODE	YEAR	MONTH_IN_YEAR	MONTHNAME
200801	2008	1	January
200802	2008	2	February
200803	2008	3	March
200804	2008	4	April

Using Aggregates

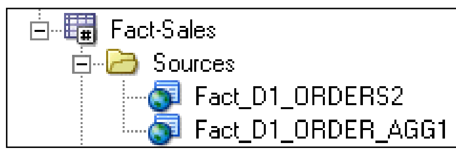
Using aggregate tables is the one of the most important strategies for boosting the query performance in the Oracle BI environment. While the need for creating and maintaining aggregates is easy to understand, using aggregates is often not effectively implemented or not implemented at all for various reasons. Because aggregates are summaries of the detailed fact tables at higher levels along the dimension hierarchies, aggregate tables have fewer rows than the base tables, resulting in improved query performance when Oracle BI Server uses the aggregate (smaller) table to satisfy the query. Because the fact table can join with multiple dimension tables, it is possible to create aggregates at a variety of levels. Creating the right amount of aggregates with appropriate levels is a trial and error exercise that requires you to analyze the nature of queries, including how results are reported and what levels. Having too many aggregates is as bad as having none. Typically, the aggregate tables should compress the detailed fact table data by a significant degree and should not result in significant storage need or process need to populate the aggregate tables.

Using Aggregate Navigation

Improves performance by enabling Oracle BI Server to transparently intercept queries and rewrite them to optimized data sources



Use the logical table source Content tab to define aggregation content.



Oracle BI Server chooses the most economical source when there are several sources from which to choose.

Using Aggregate Navigation

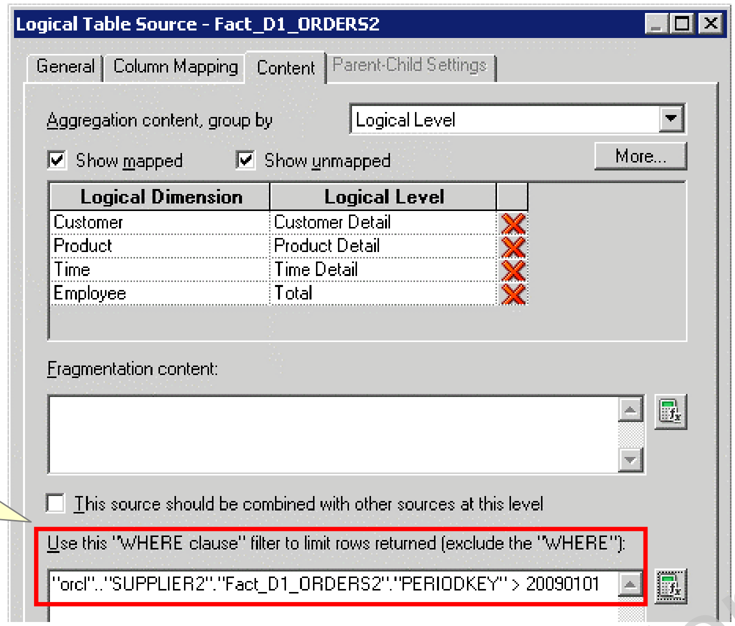
Aggregate summarization is the process of aggregating and storing records into smaller and more efficient data structures optimized for query processing. Performance gains can be realized with the right aggregate summarization deployment by reducing the time it takes to process any given query request. Aggregate navigation is the ability of Oracle BI Server to transparently intercept queries and rewrite them to optimized data sources built through an aggregate summarization process.

Every fact logical table source needs to be defined with aggregation content to ensure that Oracle BI Server chooses the most economical source when there are several sources to choose from. The aggregation content allows Oracle BI Server to pick the aggregate table (if there is one) instead of the detailed fact table without the need to explicitly specify the table name in the query. If there is more than one logical table source with identical aggregation contents, the most economical source is determined by the logical size of the fact table source, which is determined by the combined “number of elements at this level” property of the levels of the aggregate content. If a logical table source does not contain aggregation content, Oracle BI Server assumes that the logical table source is at the lowest (detailed) level. In that case, it is good practice to define aggregation content for the logical tables source at the detail level. Although it is possible to define the aggregation content by logical level or by column, you should always define the aggregation content by logical level.

Constraining Results by Using a `WHERE` Clause

Improves performance by limiting the rows returned from a data source

Use the `WHERE` clause filter on the Content tab of the logical table source to limit the rows returned from the database.

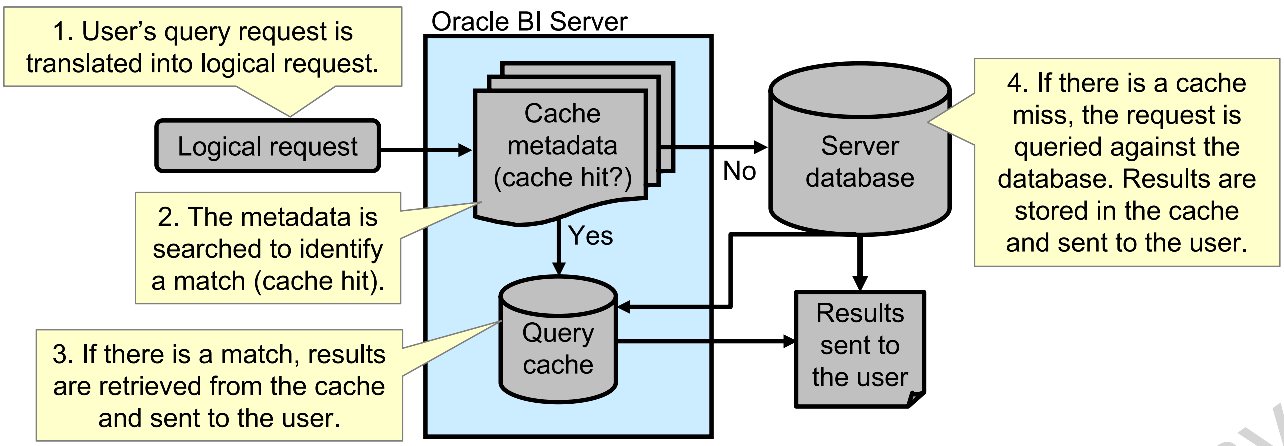


Constraining Results by Using a `WHERE` Clause

If a logical table source contains unwanted data or data that you would not typically use in a report, use the `WHERE` clause filter on the Content tab of the logical table source to limit the rows returned from the database. Because constraints in the `WHERE` clause are made on the physical tables on the source, it is helpful if indexes are defined on the columns that are used as constraints in the `WHERE` clause.

Caching

Improves performance by fulfilling a query from a local cache as opposed to processing the query through a data source



Caching

Caching is one of the most important strategies you can use to improve query performance and reduce computing resource demand of Oracle BI Server. Caching eliminates unnecessary database processing because pre-computed results are stored in a local cache. It improves query performance by fulfilling a query from the cache, as opposed to searching through the database. It also conserves network resources by avoiding a connection to the database server.

Caching does not guarantee a solution to all performance problems; it should therefore not be relied on to overcome the limitations of a poorly designed repository. Caching is meant for Oracle BI Server to reuse cache results from a large number of small queries. Caching should be enabled and configured in the production environment. For best performance, the cache storage directories should reside on the local disk or dedicated drives. Set the maximum number of rows for any cache entry (`MAX_ROWS_PER_CACHE_ENTRY`) and the maximum number of cache entries (`MAX_CACHE_ENTRIES`) parameters to avoid using up the cache space with runaway queries that return large number of rows. Cache management strategies should be in place, including pre-populating the cache using cache-seeding techniques (such as iBots) provided by Oracle BI Server and keeping the cache up to date using cache-purging techniques.

Limiting the Number of Initialization Blocks

Improves performance because initialization block queries are executed when Oracle BI Server is started and when users log in to the server

```
-------------------- An initialization block named
'CurrentPeriods', on behalf of a Repository Variable, issued the
following SQL query:

    SELECT YYYYMMDD, MONTHCODE, YEAR FROM SUPPLIER2.D1_CALENDAR2
WHERE YYYYMMDD = (SELECT MAX(PERIODKEY) FROM
SUPPLIER2.D1_ORDERS2)

Returned 1 rows.  Query status: Successful Completion
```

ORACLE

Limiting the Number of Initialization Blocks

Initialization blocks are the only means to initialize dynamic repository, system session, and non-system session variables, but care should be taken not to create too many of them. In the case of dynamic repository variables, the SQL in the initialization blocks is executed every time the server is started or periodically if a schedule is set up to refresh the value of the variable.
In the case of system and non-system session variables, the initialization blocks are executed every time a user logs in to the server.

Initialization blocks for repository variables can also have a refresh interval that adds a recurrent overhead.

Limiting Select Table Types

Improves performance because it:

- Reduces the number of `SELECT` statements executed by Oracle BI Server
- May avoid lengthy SQL queries

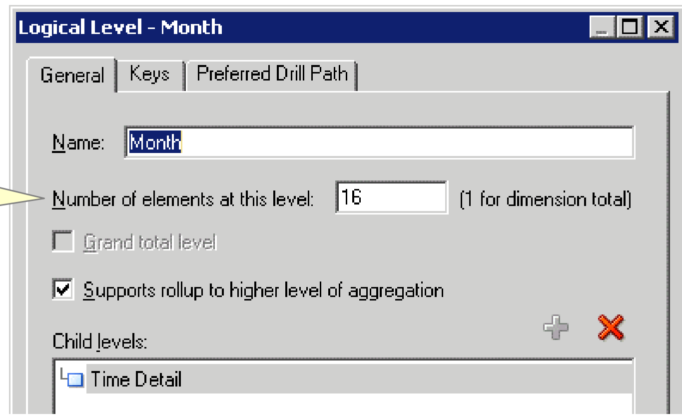
Limiting Select Table Types

`Select` table types (opaque views) in the Physical layer can perform a vital function in some cases. However, when defining a table as a `Select` type, avoid long SQL statements that may affect performance. It is also a good idea to limit the number of `Select` tables in the Physical layer because the associated `SELECT` statement is executed every time by the server, which could affect overall performance. The `Select` table type can be replaced with a physical table or view in the database.

Modeling Logical Dimension Hierarchies Correctly

Improves performance by ensuring that Oracle BI Server chooses the most economical source

With multiple aggregate sources, specify the number of elements for each level.



The number does not have to be exact, but ratios of numbers from one parent to child logical level should be accurate.

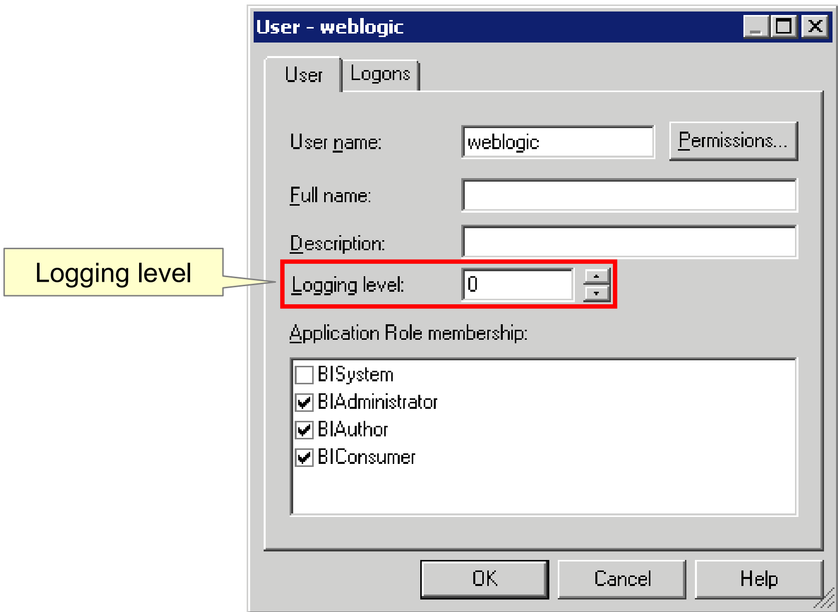
Modeling Logical Dimension Hierarchies Correctly

Logical dimension hierarchies must be modeled accurately to ensure that the Oracle BI optimizer chooses the most economical source. When there are multiple aggregate logical table sources, the number of elements for each level must be specified. The number does not have to be exact, but ratios of number from one parent to child logical level should be accurate.

All levels except the Grand Total level need to be defined with at least one column from the dimension table. However, it is not necessary to explicitly associate all the columns from the dimension table with logical levels. Columns that are not associated with a logical level are automatically associated with the lowest level in the dimension that corresponds to that dimension table.

Turning Off Logging

Improves performance in a production environment because Oracle BI Server does not generate log files



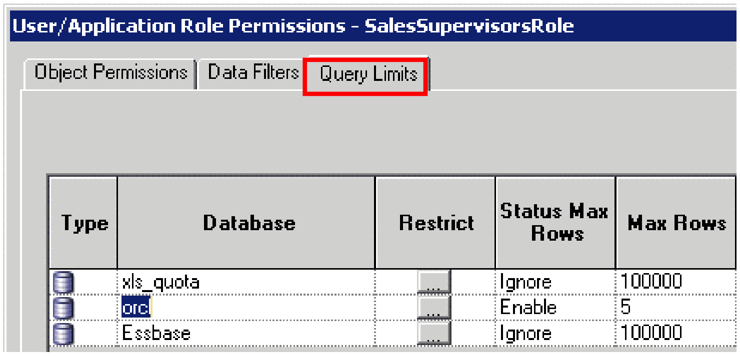
Turning Off Logging

The logging facility is disabled by default because it can affect the performance of Oracle BI Server and can create large log files. Although logging can be used to test and troubleshoot problematic queries, it should not be enabled in a production environment, especially for the Administrator user, because repository variable initialization blocks are logged to the Administrator.

If you decide to enable logging, consider creating a separate Administrator user with a logging level of 2, which allows the user to debug using the query log and should provide enough logging information in most cases. Logging levels greater than 2 should not be used because of the possibility of a severe impact on query performance in a large user environment. Logging levels greater than 2 are typically used only by Oracle technical support.

Setting Query Limits

Improves performance by enabling Oracle BI Server to track and cancel runaway queries



Setting Query Limits

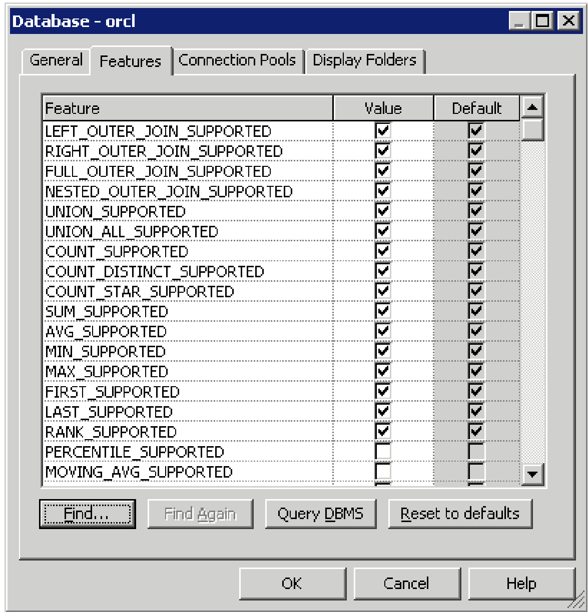
In addition to limiting the maximum number of rows being returned by Oracle BI Presentation Services, you can also enable Oracle BI Server to track and cancel runaway queries by placing various limits on the repository for a given user or application role. For each user or application role, it is possible to limit queries by varying conditions:

- Maximum number of rows that a query can retrieve from a database
- Maximum time that a query can run on a database
- Restricting access to a database during particular time periods from Oracle BI Server

Note that limits can be set by user or application role and by database. Oracle recommends setting limits by application role. Note that with some "databases" such as Excel, more processing may have to be done by Oracle BI Server because fewer operations can be function-shipped to the database. This means that limits should not be made too low.

Pushing Calculations to the Database

Improves performance by automatically pushing certain operations to the database based on database feature entries



Pushing Calculations to the Database

Oracle BI Server is intelligently designed to take maximum advantage of a database's processing resources when it knows that the database can process certain operations more efficiently than it can process the same operations. Oracle BI Server automatically determines this based on database feature table entries, compensates for the database's lack of functionality, and pulls back appropriate data for postprocessing before returning the result set to the user. By modeling the repository accurately and adjusting the default database feature table entries for a database, you should be able to achieve a combined optimal performance of both Oracle BI Server and the database.

Exposing Materialized Views in the Physical Layer

Improves performance because exposing materialized views explicitly guarantees that Oracle BI Server chooses the most economical table source to satisfy a query

Exposing Materialized Views in the Physical Layer

Materialized views are schema objects that are used to compute and store aggregated data such as sums and averages. If materialized views are used in the database, they should be exposed in the Physical layer of the repository and modeled as regular aggregate tables. Because materialized views are hidden objects, Oracle BI Server has to rely on the query optimizer of the database for query rewrites to use correct materialized views. Exposing them explicitly guarantees that Oracle BI Server chooses the most economical table source to satisfy a query.

Using Database Hints

Improves performance because hints force the database query optimizer to execute the statement in a more efficient way

Using Database Hints

Hints are instructions placed within a SQL statement that tell the data source query optimizer the most efficient way to execute the statement. Hints override the optimizer's execution plan, so you can use hints to improve performance by forcing the optimizer to use a more efficient plan. Hints are supported for Oracle Database data sources only.

Using the Administration Tool, you can add hints to a repository, in both online and offline modes, to optimize the performance of queries. When you add a hint to the repository, you associate it with Physical layer objects. When the object associated with the hint is queried, Oracle BI Server inserts the hint into the SQL statement.

Hints that are well researched and planned can result in significantly better query performance. However, hints can also negatively affect performance if they result in a suboptimal execution plan. Use the following guidelines to create hints to optimize query performance:

- Add hints to a repository only after you have tried to improve performance in the following ways:
  - Added physical indexes (or other physical changes) to the Oracle database
  - Made modeling changes within the server
- Avoid creating hints for physical table and join objects that are queried often. If you drop or rename a physical object that is associated with a hint, you must alter the hints accordingly.

Summary

In this appendix, you should have learned how to identify techniques to optimize Oracle BI query performance.

Summary

Consult the Oracle Business Intelligence documentation for more information about the topics covered in this appendix.

B

Model First Development Methodology

Model First Development Methodology: Overview

- Recommended approach for developing Oracle BI repositories
- Driven by business analysis and usage history
- Iterative, top-down approach that focuses on the consolidation and abstraction of core business requirements irrespective of the underlying physical architecture:
  - Build business model first.
  - Integrate with underlying physical architecture.
  - Quickly deploy baseline to end users.
  - Pursue iterative development based on user feedback.

Model First Development Methodology: Overview

Developers using a top-down, model first approach start with the business requirements, prioritize actual business use cases, and then methodically drill down to the technical details.

This top-down approach is generally regarded as a macro view of the issues related to development, based on business drivers and users, and driven by business analysis. This is opposed to bottom-up analysis, which starts at the physical sources, catalogues technical details (databases, tables, columns, and so on), and then rationalizes the details into categories that attempt to match historical end-user usage patterns and business activities. Bottom-up analysis is an appropriate performance analysis technique for highly repeatable, static events, but it falls short in dynamic, ad hoc environments in which usage is less predictable. It is entirely possible for a thorough performance optimization process driven by a bottom-up analysis to become obsolete within days or weeks as the business adapts to differing views of the data.

The main objective of the model first development approach is to build the desired business model first. This ensures that functional and inherent performance requirements are addressed by the logical integration and augmentation of the physical architecture. This approach enables developers to quickly get a reasonable baseline business model in front of users, and then pursue iterative development based on user feedback.

Central Tenets of the Model First Development Methodology

- Rapid prototyping
  - Leverage actual subsets or fictitious physical data stores and manageable data volumes to reduce performance issues during development.
- Iterative development and user feedback
  - Leverage prototypes for demos and hands-on sandboxes.
  - Rollout augmented models frequently.
  - Demonstrate responsiveness to feedback.
- Gap analysis
  - Map the business model to actual physical sources.
  - Manage scope and expectations accordingly.
- Gather performance requirements along the way
  - Identify patterns of use, data granularity, user groups, and security constraints.

Central Tenets of Model First Development Methodology

A model first methodology should employ rapid prototyping of the repository business model by leveraging actual subsets or fictitious physical data sources and manageable data volumes. This helps eliminate performance issues during development. After a prototype is built and delivered to users, iterative development should continue based on user feedback. This iterative development should ensure that augmented business models are rolled out frequently and that they demonstrate responsiveness to user feedback. During this process, developers must perform gap analysis to map the business model to actual physical data sources. Since there will be gaps, it is important to manage project scope and expectations accordingly. Developers should also gather performance requirements along the way by identifying patterns of use, data granularity, user roles and groups, and security constraints.

Baseline Performance Analysis

Enables you to analyze performance requirements better and discuss them with business users:

- Catalog and enumerate all users, roles, job titles, and objectives.
- Obtain a detailed organization chart of the business intelligence (BI) users and organizations.
- Identify mandated security model guidelines.
- Collect row counts for all tables (facts and dimensions) as well as dimension-level cardinality.

Baseline Performance Analysis

It is important to perform a baseline performance analysis. The output of a top-down performance optimization analysis is a rationalized and prioritized framework of dimensions, levels, and measures from which to build and map aggregate summary objects. Upfront analysis of the key data points in the slide prepares you to discuss performance requirements with users better.

From these four straightforward data inputs, it is often possible to pull together a workable framework that can be implemented across large user bases. Natural groupings become apparent from the user, role, title, and objectives profile. Groups are easily organized via similar objectives. Key dimensions and levels can quickly surface.

Organization charts, when held up to user groups, often overtly expose key dimension and levels such as geographic or product alignments and provide candidates for summaries.

Baseline Performance Analysis (continued)

Security requirements show the required dimensionality for record filtering and can even eliminate entire groups from aggregate summary analysis due to highly restrictive data access requirements.

Records counts and cardinality are instrumental in calculating record compression and validating aggregate summary prototypes. Without significant compression, the aggregate summaries may prove too time-intensive to build or not worth the effort.

Defining the Business Model: Dimensional Matrix

Create a dimensional matrix that places business processes (facts) along one axis and dimensionality along the other.

Business processes

	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5
Fact 1					
Fact 2					
Fact 3					
Fact 4					
Fact 5					

ORACLE

Defining the Business Model: Dimensional Matrix

One of the most usable tools in rationalizing BI users' requirements is a dimensional matrix. This is a simple two-dimensional matrix that places business processes (facts) along one axis and dimensionality along the other. It provides a visual medium to ground discussions. Before beginning development, you gather business requirements in terms of dimensions and measures, identify business processes, and determine key performance indicators. The output of a top-down analysis is a rationalized and prioritized framework of dimensions, levels, and measures from which to build and map aggregate summary objects.

Dimensional Matrix: Example

A business model is a completed matrix that resembles the following graphic, with X denoting key dimensions for a given business process and O denoting minor dimensionality.

Business processes	Time	Account	Organization	Product	Geography
Sales	X	O	X	X	X
Forecast	X		X	X	O
Service	X	X	O	X	X
Orders	X	X	O	X	O
Activities	X	X		O	O

X	Frequently
O	Sometimes
	Never

Drill to Levels for More-Detailed Performance Requirements

Each business process is individually rationalized against the dimensional hierarchies and the user roles to which they apply.

	Time				Organization				Product		Geography			
	Year	Quarter	Month	Day	Level 3 Position	Level 2 Position	Level 1 Position	Level 0 Position	Product Line	SKU	Region	Country	State	City
Sales	X	X	X	O	X	X	X	X	X	O	X	X	O	O
Forecast	X	X			X	X	X	X	X		X	X		

Sales Manager Role

Drill to Levels For More detailed Performance Requirements

Each business process is individually rationalized against the dimensional hierarchies and the user roles to which it applies. Matrix tools such as this are highly useful in keeping performance requirements focused on key aspects and ultimately providing a concise, documented design for an aggregate summary model.

Focus on the Business Model

Focus on the business model rather than the presentation:

- Ad hoc reports are typically used once and are not pervasive.
- Existing reports are useful only when abstracted for their dimension and measure objects.

Leverage Oracle BI “Legless” Applications

- Oracle BI Applications are complete, prebuilt BI solutions:
  - The prebuilt Oracle BI Applications repository contains business models that can be mapped to different physical data sources.
  - Value can be realized without Oracle Business Analytic Warehouse and ETL components.
- Redefine the Oracle BI Application Physical layer objects by using the “opaque view” feature:
  - Use `SELECT` statements.
  - Deploy opaque views via BI Server Administrator as necessary.
  - Materialize as required.

Leverage Oracle BI “Legless” Applications

Oracle Business Intelligence Applications are complete, prebuilt BI solutions that deliver intuitive, role-based intelligence for everyone in an organization to enable better decisions, actions, and business processes. Based on best practices, these solutions enable you to gain greater insight and value from a range of data sources and applications, including Oracle E-Business Suite, PeopleSoft, Siebel, and third-party systems such as SAP.

Oracle BI Applications are built on the Oracle BI platform. Thus, the prebuilt Oracle BI Applications repository contains prebuilt business models that can be mapped to different physical data sources. These “legless” applications enable organizations to realize the value of a packaged BI application, such as rapid deployment, lower total cost of ownership (TCO), and built-in best practices, while also being able to extend those solutions easily to meet specific needs. You also have the option to build completely custom BI applications, as you did in this course, all on one common BI foundation.

Use Oracle BI Data Mart Automation

The Aggregate Persistence Wizard automates the aggregate summary process.

- Use this wizard-driven utility in the Administration Tool to define, populate, store, and map aggregated data stores:
  - Choose the measures that should be aggregated.
  - Choose the dimensions and levels to be aggregated to.
  - Select the data sources in which to physically store the aggregate summaries.
- Create query performance improvement over normalized, transaction-level physical schemas.

Use Oracle BI Data Mart Automation

As you saw in the lesson titled "Using Aggregates," the Aggregate Persistence Wizard automates the aggregate summary process. On completion of the wizard, a template is generated to be deployed and scheduled with the Oracle BI Scheduler or the customer's scheduling tool of preference. Aggregate summaries are the most effective and scalable way to pervasively manage end-user query performance.