Oracle Enterprise Manager
Oracle Database and Application Testing

Data Masking Lab

Session S318966

Oracle Enterprise Manager 11g
Data Masking Hands-on Lab

Introduction to Enterprise Manager 11g

Oracle Enterprise Manager 11*g* is the centerpiece of Oracle's integrated IT management strategy, which rejects the notion of management as an after-thought. At Oracle, we design manageability into each product from the start, enabling Oracle Enterprise Manager to then serve as the integrator of manageability across the entire stack encompassing Oracle and non-Oracle technologies. Fueled by this unique vision, Oracle Enterprise Manager 11*g* has introduced *business-driven IT management* to help IT deliver greater business value through three highly differentiated capabilities:

- Business-driven application management, which combines industry-leading capabilities in real user experience management, business transaction management and business service management to improve application users' productivity while enhancing business transaction availability
- Integrated application-to-disk management, which provides deep management across the entire Oracle stack to reduce IT management complexity and eliminate disparate point tools
- Integrated systems management and support, which utilizes industry-first technology bring support services into the IT management console; enabling proactive IT administration, increased application and system availability, and improved customer satisfaction

Introduction to Enterprise Manager 11g Data Masking Pack

Oracle Data Masking pack for Enterprise Manager, part of Oracle's comprehensive portfolio of database security solutions, helps organizations comply with data privacy and protection mandates such as Sarbanes-Oxley, Payment Card Industry (PCI) Data Security Standard (DSS), Health Insurance Portability and Accountability Act (HIPAA), as well as numerous laws that restrict the use of actual customer data. With Oracle Data Masking, sensitive information such as credit card or social security numbers can be replaced with realistic values, allowing production data to be safely used for development, testing, or sharing with out-source or off-shore partners for other non-production purposes.

- **Comprehensive and Extensible Mask Library** --Oracle Data Masking Pack provides a centralized library of out-of-the-box mask formats for common types of sensitive data, such as credit card numbers, phone numbers, national identifiers..
- **Sensitive Data Discovery and Application Integrity** -- Using Oracle Data Masking Pack's search capabilities, information security administrators can quickly search the database to identify sensitive data. In some applications, the same sensitive data is maintained in multiple tables related by referential (primary key-foreign key) relationships. Oracle Data Masking Pack discovers these relationships and masks all related data elements automatically while preserving referential relationships.
- **Sophisticated Masking Techniques** -- Oracle Data Masking Pack provides a variety of sophisticated masking techniques to meet application requirements while ensuring data privacy: **Condition-based** masking which makes it possible to apply different mask formats to the same data set depending on the rows that match the conditions, **Compound** masking which ensures that a set of related columns is masked as a group to ensure that the masked data across the related columns retain the same relationship, and **Deterministic** masking which ensures repeatable masked values after a mask run. Enterprise may use this technique to ensure that certain values get masked to the same value across all databases.

- **Secure High Performance Mask Execution** -- Unlike traditional masking processes that are typically slow, Oracle Data Masking Pack uses highly efficient parallelized bulk operations to replace the original sensitive data with masked data. Because the entire data masking process is done in place, enterprises can be assured of a greater sense of security knowing that the sensitive data would never leave the database during the masking process.
- **Support for Heterogeneous Databases**: Oracle Data Masking Pack can support masking of data in heterogeneous databases, such as IBM DB2 and Microsoft SQLServer, through the use of Oracle Database Gateways.

This lab will demonstrate:

- Creating and exporting data masking formats
- Masking sensitive application data
- Using compound masking, condition-based masking and user defined masking
- (**OPTIONAL**)Deterministic masking

Please feel free to seek assistance from the instructor or Oracle Demo staff at any point in time.

Before we start taking you through the demonstration, please note the following:

- You will be given a virtual machine address to use for this lab. For ease of reference, you may want to write this below:

Virtual Machine Address: _____

- You will connect to that system using VNC. VNC password is g0Oracle12#
- Operating System Accounts: oracle/g0Oracle12# and root/g0Oracle12#
- Database(db04 and db05) Accounts: system/oracle1
- Grid Control Accounts: sysman/oracle1

Additional information can be found at:
Demo Booths located at {Location}
Additional Sessions:
Moscone South: Enterprise Manager # XXXX
Moscone West: Enterprise Manager # XXXX

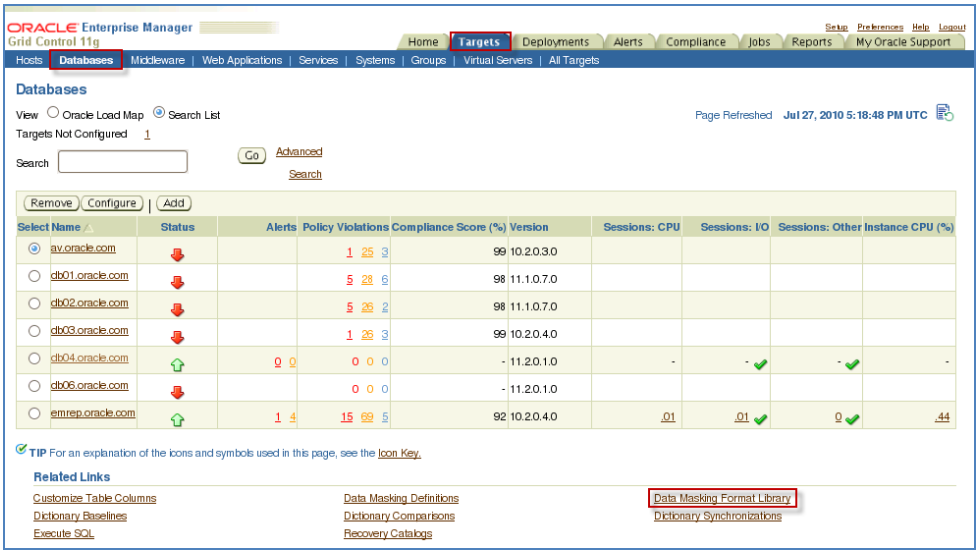
For additional information, visit:

Oracle Enterprise Manager
http://www.oracle.com/enterprise_manager/index.html

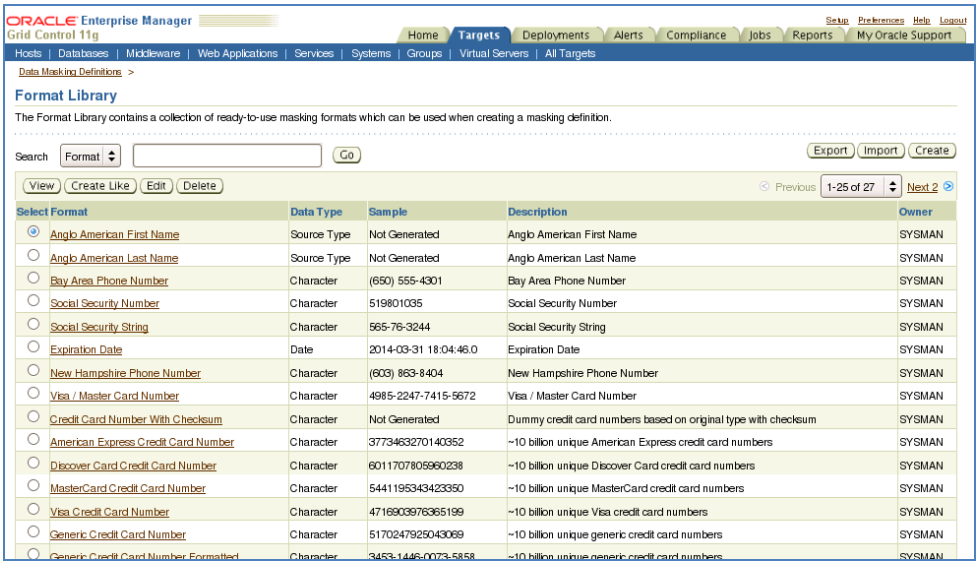
Creating and exporting data masking formats

1. Start Firefox and login to Grid Control as **sysman/oracle1** at the URL **http://dbsecurity.oracle.com:4889/em**.

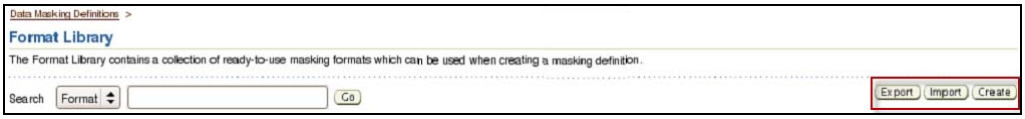
2. Navigate to **TARGETS**->**DATABASES**-> **Data Masking Format Library**



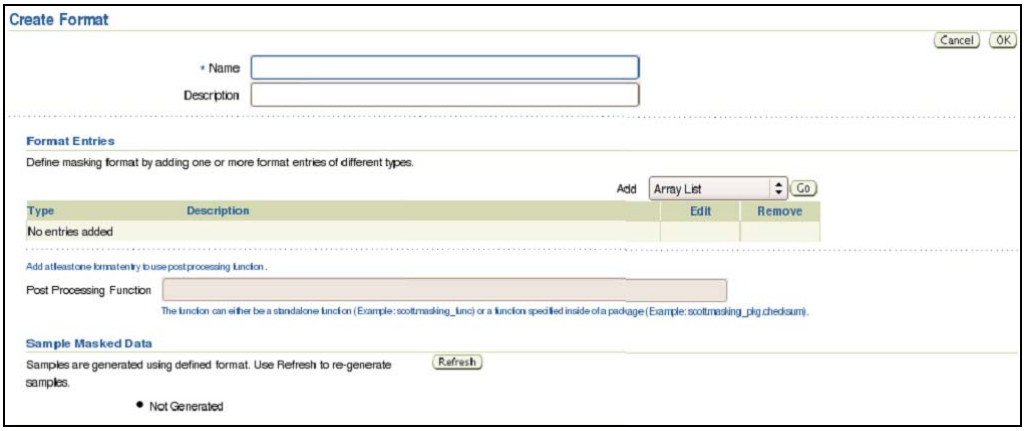
3. The format library contains a collection of ready-to-use masking formats. The library consists of format routines that you can use for masking. A masking format can either be one that you create, or one from the list of Oracle-supplied default masking formats.



4. Click on the Create button to begin creating a custom Masking Format.

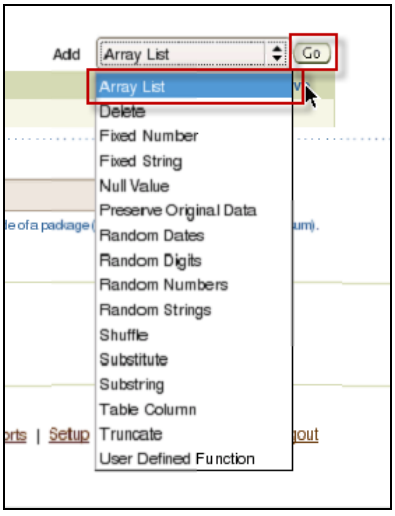


5. From the Create Format Dialog, we will configure our Masking Format

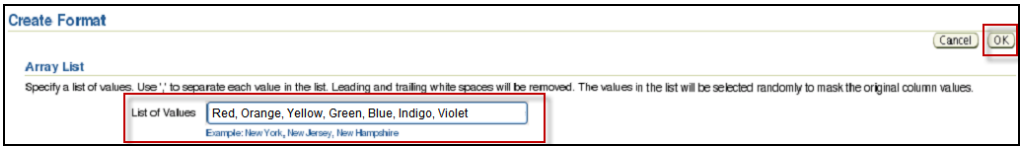


Name: Colors
Description: Colors of a rainbow

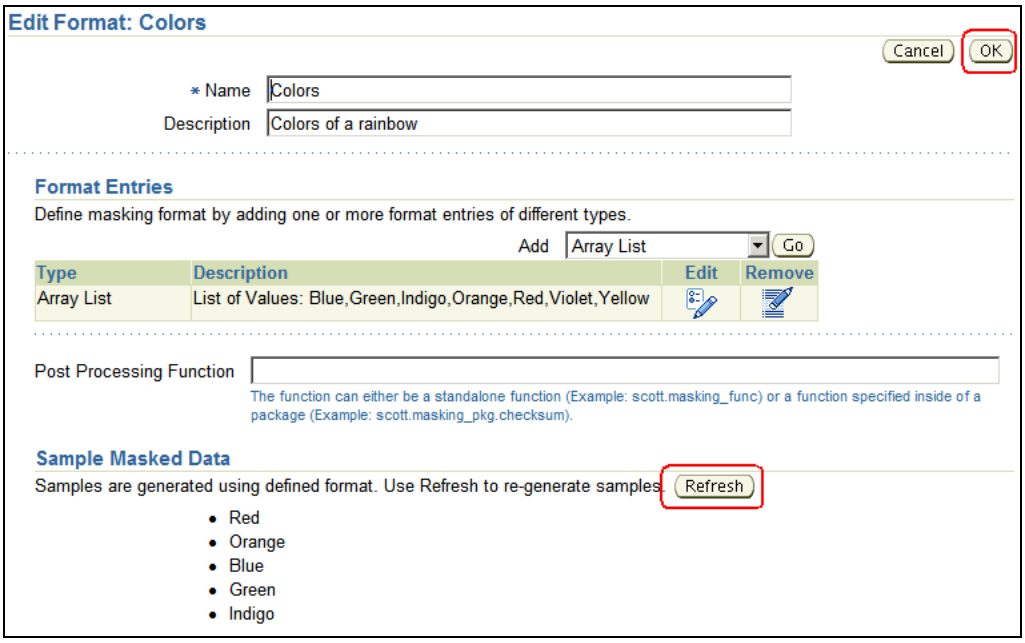
6. Type 'Colors' in the Name field and 'Colors of a rainbow' in the Description Field. Before adding a field type, view the number of different options which you can mask data. Choose Array List and click the Go button.



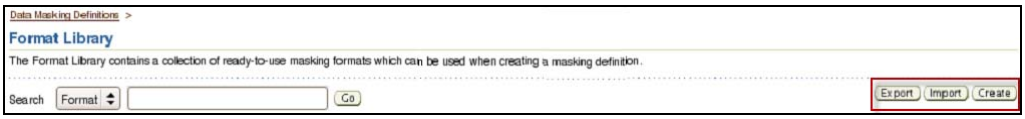
7. Define the List of Values for the Colors Format and click on the OK button when finished. The values include:
   - Red, Orange, Yellow, Green, Blue, Indigo, Violet



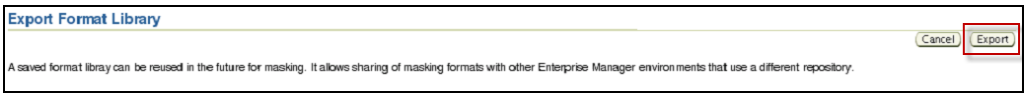
8. You can see samples of the masked data in the Sample Masked Data Section. Click on the Refresh button to see a random sample from the defined Array List. This screen allows you to edit any values of the Masking Format. Click the OK button when you are satisfied with the entries.



9. Return to the Format Library screen and click on the Export button to begin the process of exporting the entire library.

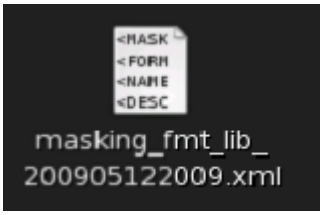


10. As the dialog states, exporting a format mask can be saved and re-used in the future for masking. This mask can be shared and/or imported into another Format Library in another Enterprise Manager environment.

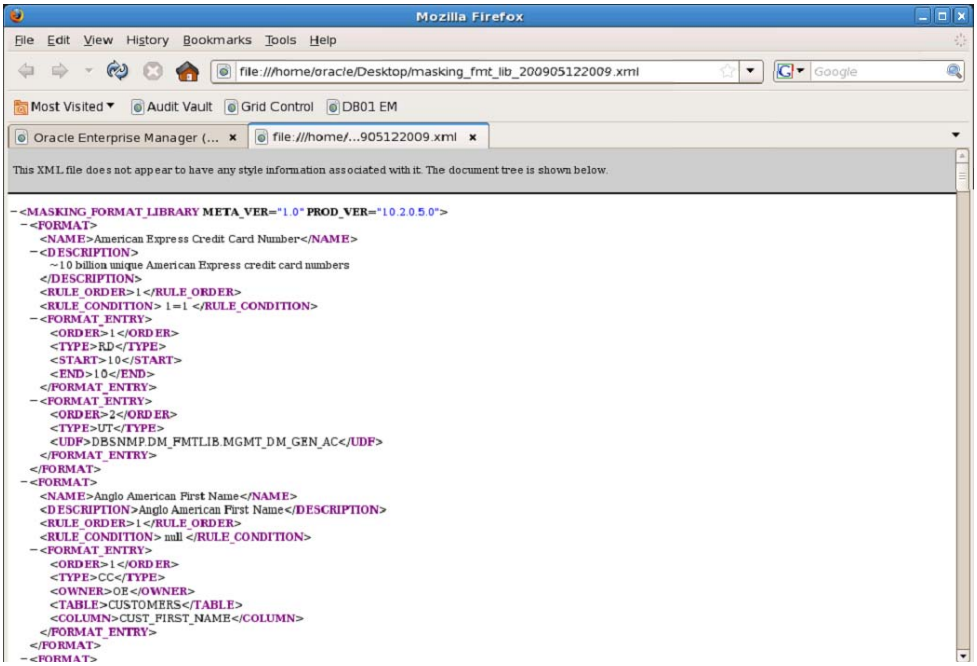


11. Save the file to the default location on the Desktop.

12. Navigate to the Desktop and double-click on the newly created XML document. Your filename will be different than what has been captured here.

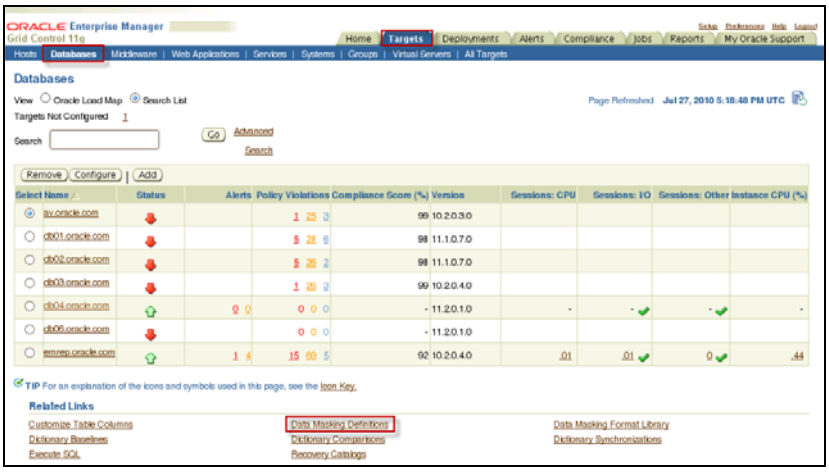


13. If you would like to, you can review the XML document and the information that has been captured in the document.

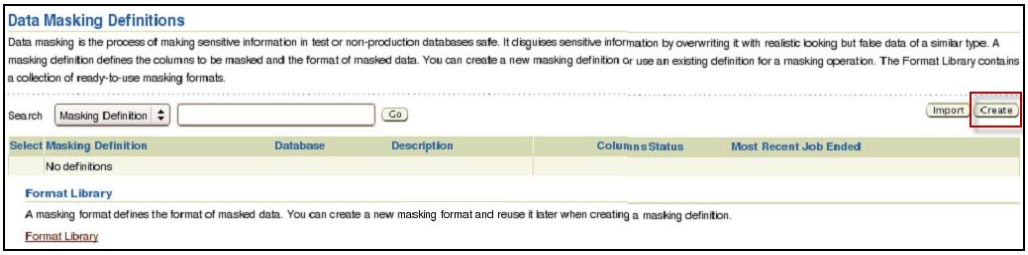


Masking Sensitive Application Data

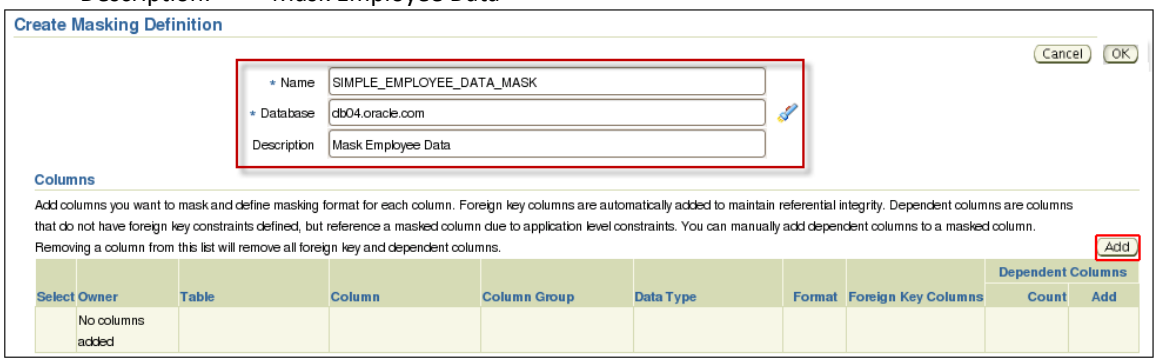
1. Navigate to the Data Masking Definitions by selecting **Targets** -> **Databases** -> **Data Masking Definitions**.



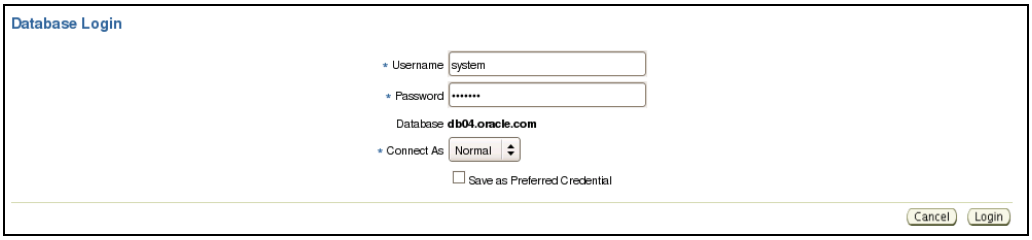
2. From the Data Masking Definitions Dialog, we will create a new definition. Click on the Create button to begin the process of masking data.



3. From the Create Masking Definition screen, type in the Name, Database and Description field with the provided values below. Continue and click on the Add button.
   i. Name: SIMPLE_EMPLOYEE_DATA_MASK
   Database: db04.oracle.com
   Description: Mask Employee Data

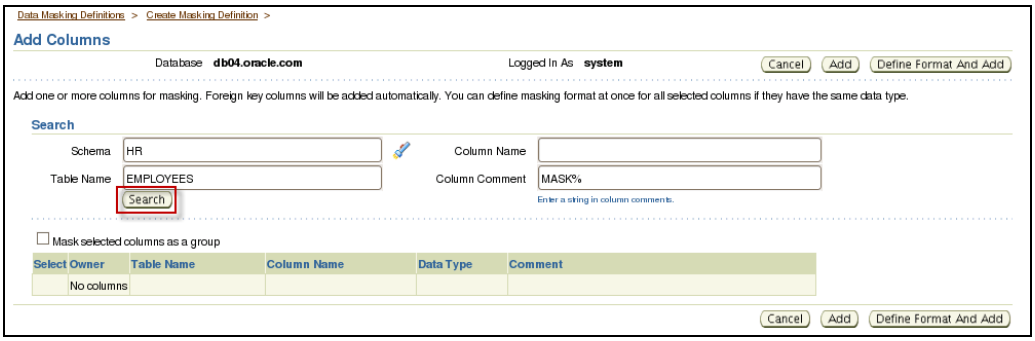


4. At the Database Login screen, login as system/oracle1. Leave “Connect As” set to Normal, and then click the Login button.

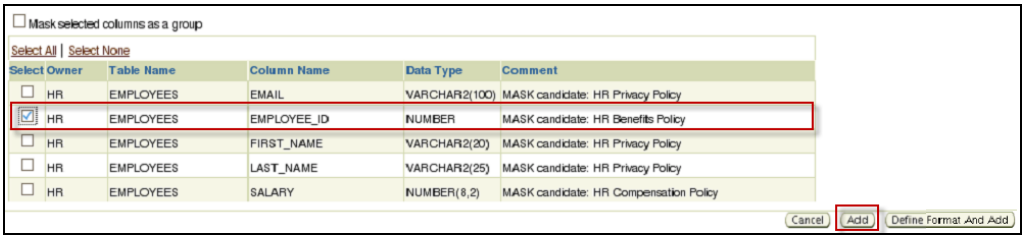


5. We are going to search for the EMPLOYEES table in the HR Schema. Type in the following values and click on the Search button.

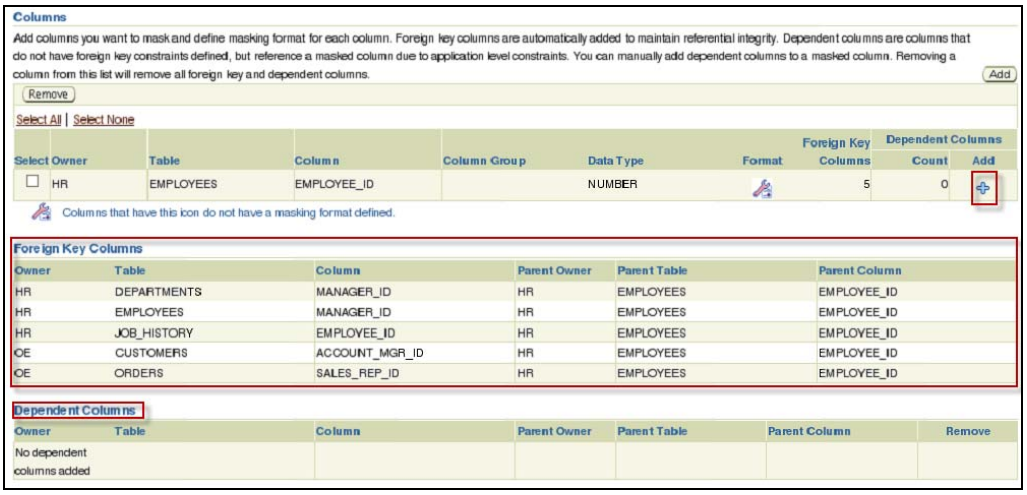
Schema:	HR
Table Name:	EMPLOYEE
Column Comment:	MASK%



6. Select the column for EMPLOYEE_ID and click the Add button.



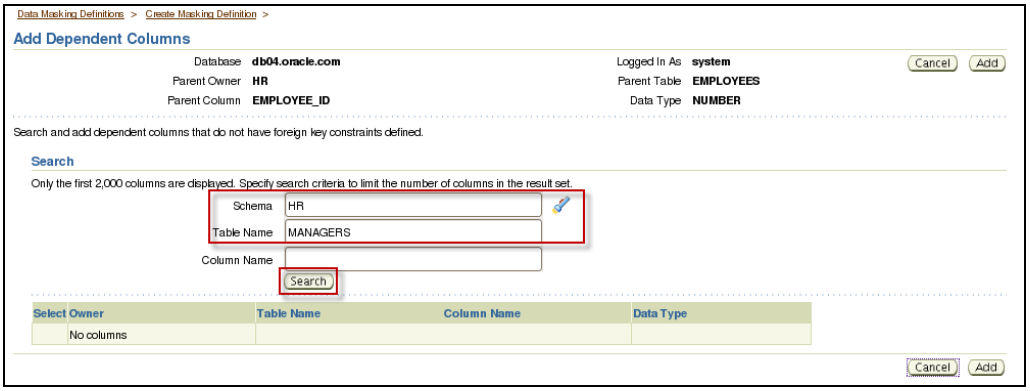
7. Notice how all associated foreign key columns (5) were added automatically to this Masking Definition. However, in this particular case, there is an additional table named MANAGERS that is part of the HR application, but all of its constraints are enforced by the application and NOT in the database. The MANAGERS table uses EMPLOYEE_ID, but the relationship is not registered in the database as a foreign key constraint. Therefore, we must add a Dependent column on the EMPLOYEE_ID column. Click on the  icon to add this Dependent Column



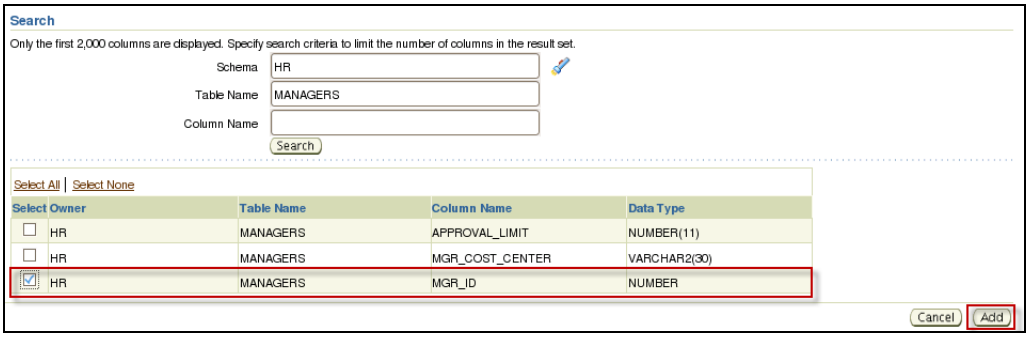
8. Type in 'HR' in the Schema and 'Managers' in the Table Name to search for the appropriate column of data. Click on the Search button to execute.

   Schema: HR

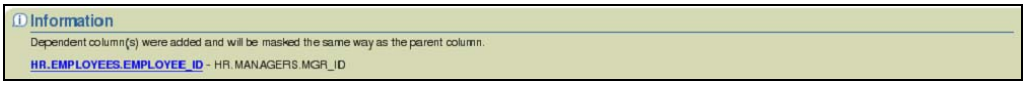
   Table Name: MANAGERS



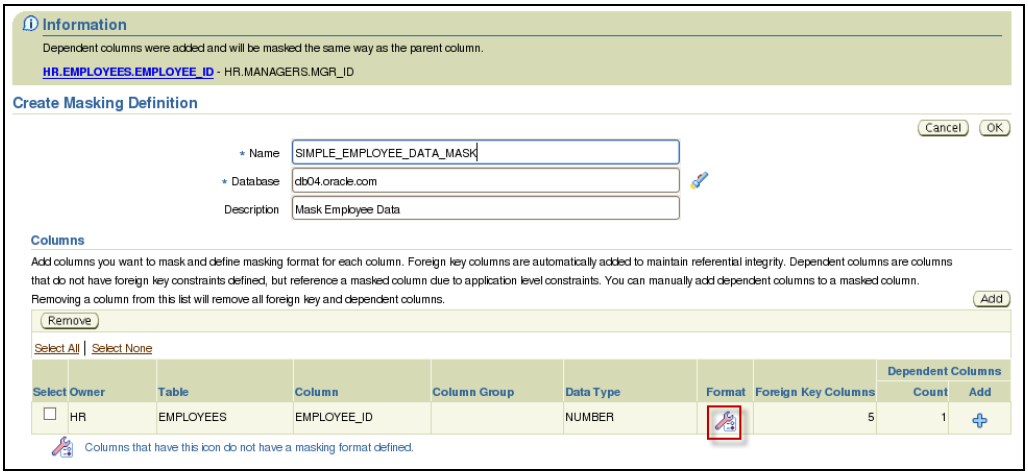
9. Select the MGR_ID column and click on the Add button.



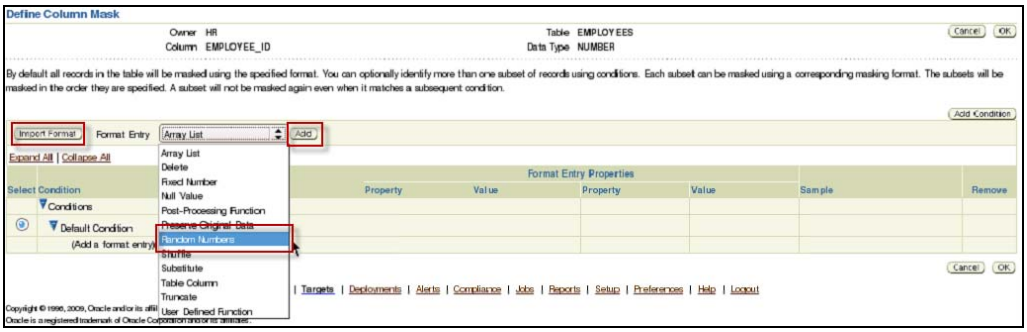
10. You have successfully added a dependent column. The dependent column HR.MANAGERS.MGR_ID will now be masked in the same way as the parent column, HR.EMPLOYEES.EMPLOYEE_ID.



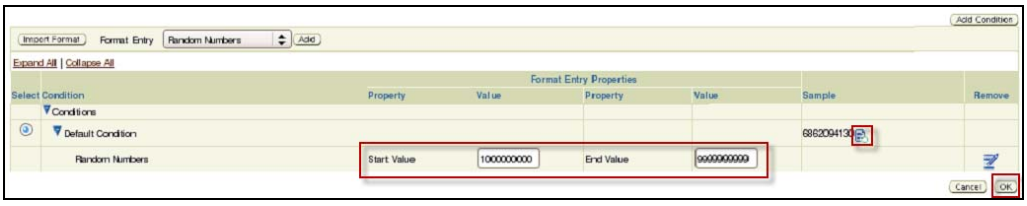
11. The next step is to format the EMPLOYEE_ID column. Continue by clicking on the  icon.



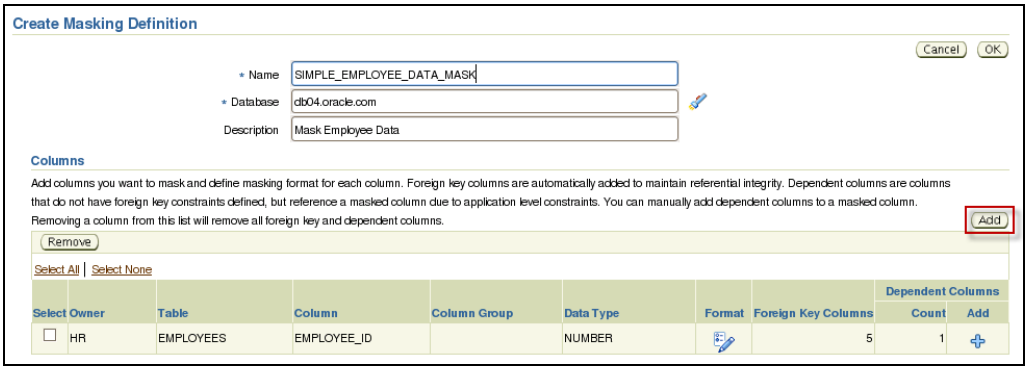
12. As previously discussed, there are many different options to format the column of data to ensure the quality of the data masking. If you were to use an existing format from the Format Library, you would click on the Import Format button. In this particular example, we are going to select Random Numbers from the drop down list box and click on the Add button.



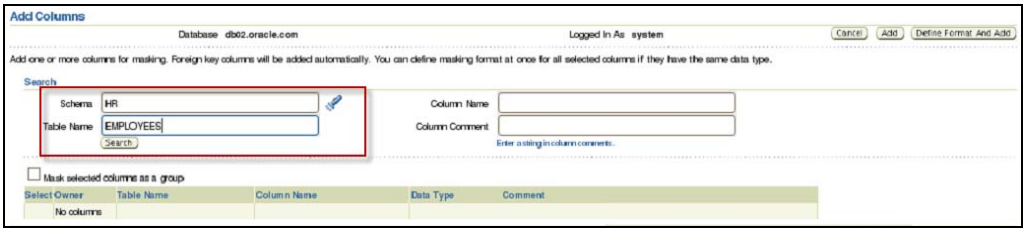
13. Enter 1000000000 for the Start Value and 9999999999 for the End Value. Click on the Sample icon  to view sample data and continue by clicking the OK button.



14. The next step is to add additional columns in the EMPLOYEES table to include in this masking operation. Click the Add button to continue.

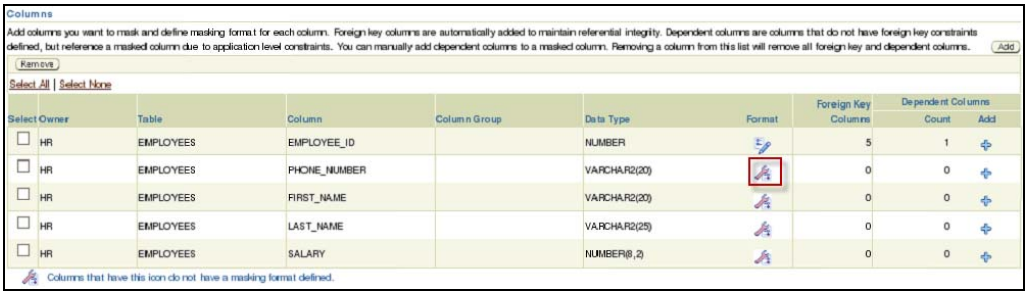


15. Set HR as the Schema and EMPLOYEES as the Table Name and click on the Search button to query for appropriate columns.



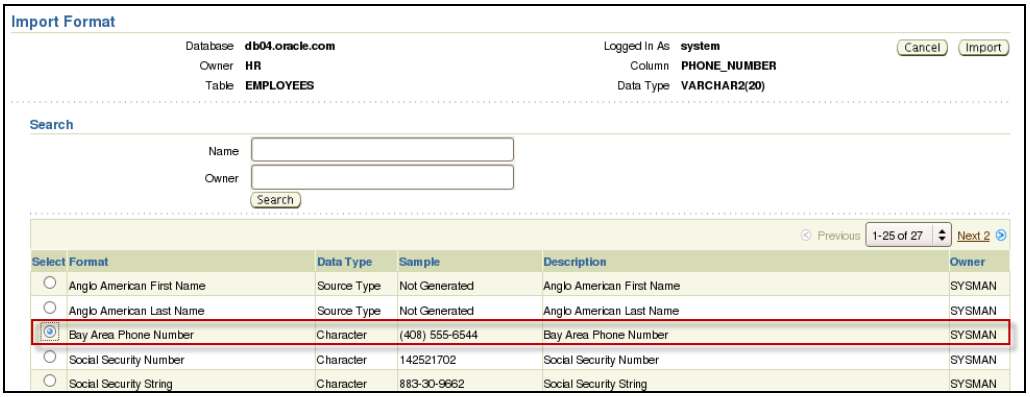
16. Add 4 columns in HR.EMPLOYEES for masking (FIRST_NAME, LAST_NAME, PHONE_NUMBER, SALARY). Select the 4 columns listed in the previous step and click on the Add button.

17. Now that we've added 4 more columns to mask, we need to define a masking format for each column. Click on the icon to define a masking format for the column PHONE_NUMBER.

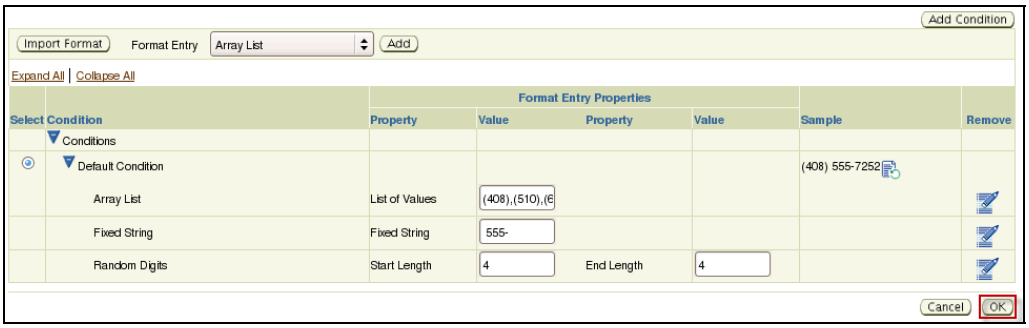


18. For the column PHONE_NUMBER, click on the Import Format button.

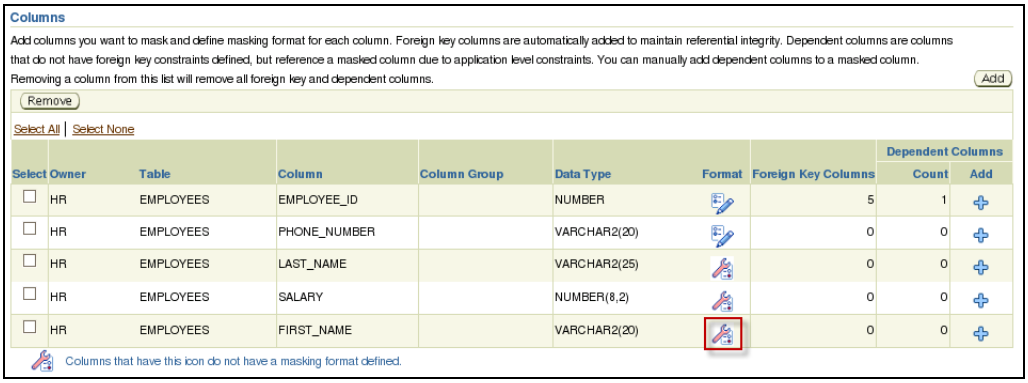
19. From the Import Format dialog, select Bay Area Phone Number and click on the Import button.



20. Review the Default Condition for the format masking for the PHONE_NUMBER column. Click on the icon to review sample data from this format mask. Click on the OK button to continue.

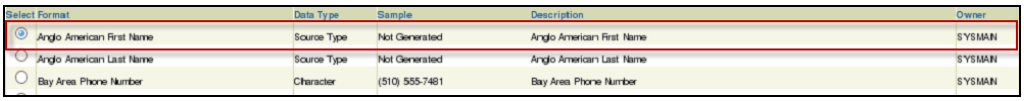


21. Continue by clicking on the icon to define a masking format for the column FIRST_NAME.

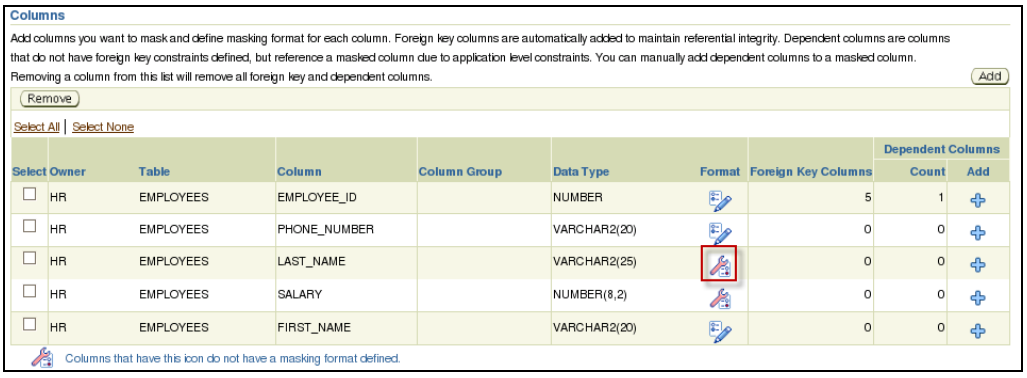


22. For the column FIRST_NAME, click on the Import Format button.

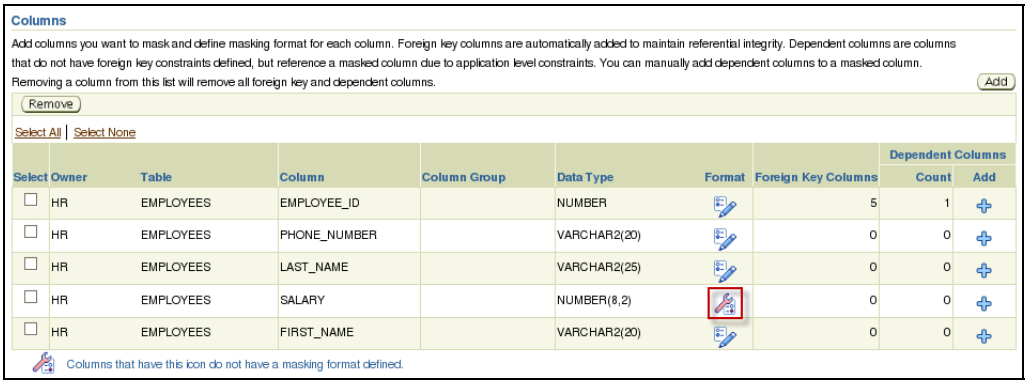
23. From the Import Format dialog, select Anglo American First Name and click on the Import button.



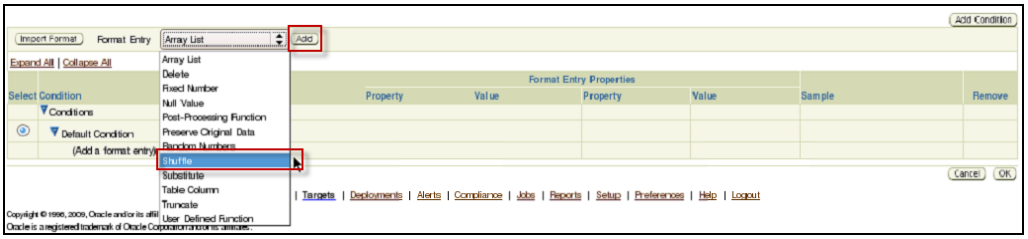
24. Repeat steps for column LAST_NAME and select the format mask Anglo American Last Name



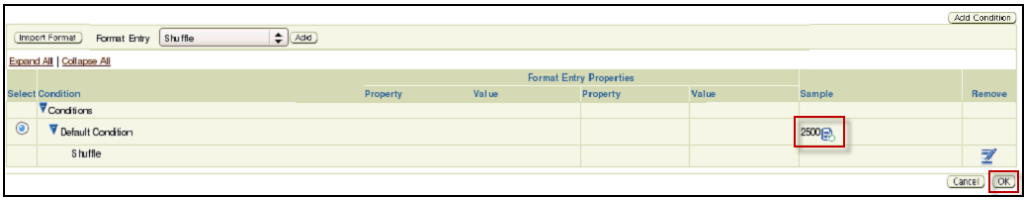
25. Continue by clicking on the icon to define a masking format for the column SALARY.



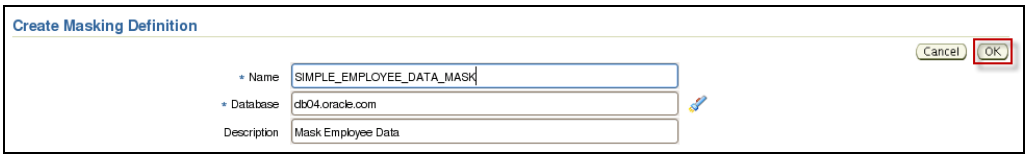
26. For this column, we will randomly Shuffle the original column data within the table. Select Shuffle from the drop-down list box and then click on the Add button.



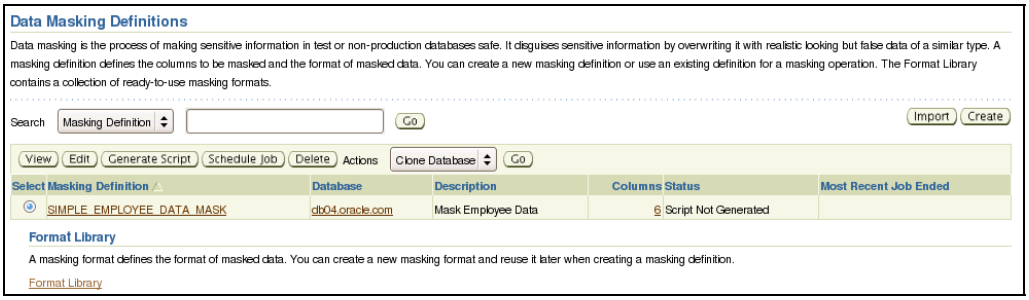
27. Review the Default Condition for the format masking for the SALARY column. Click on the icon to review sample data from this format mask. Click on the OK button to continue.



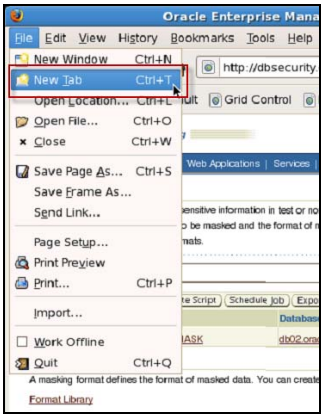
28. Click on the OK button to complete the creation of a Masking Definition for the EMPLOYEES table.



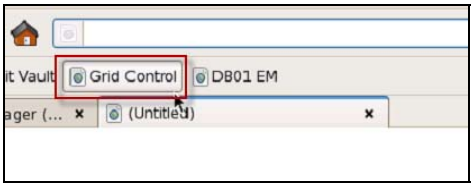
29. Review that you have now successfully created a Data Masking Definition.



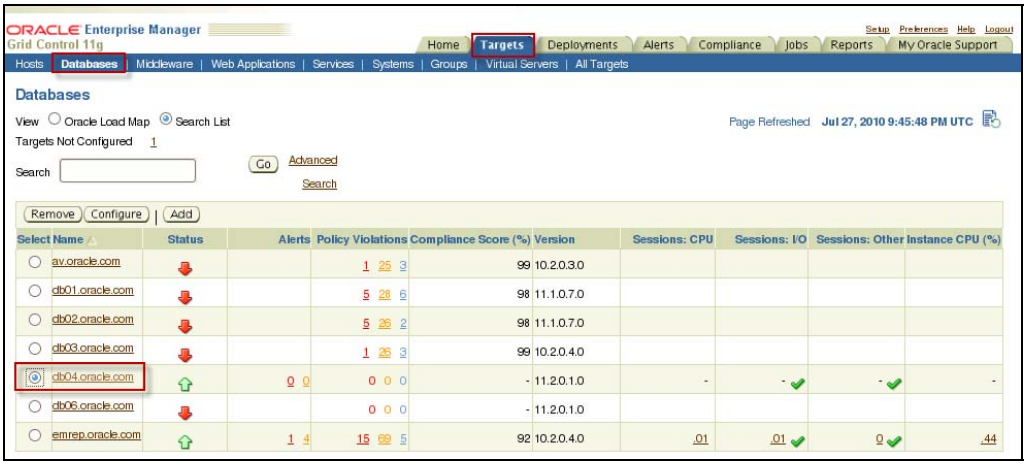
30. Before we Generate the Script to mask data, let's first query the existing unmasked data to compare the results after we mask the data. In the browser, select File -> New Tab.



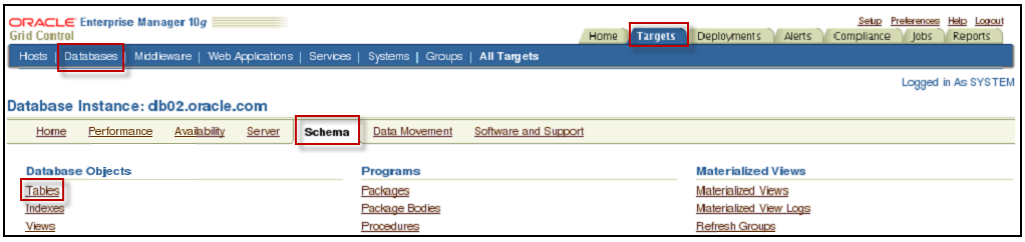
31. In the new tab, click on the shortcut to go to Enterprise Manager – Grid Control.



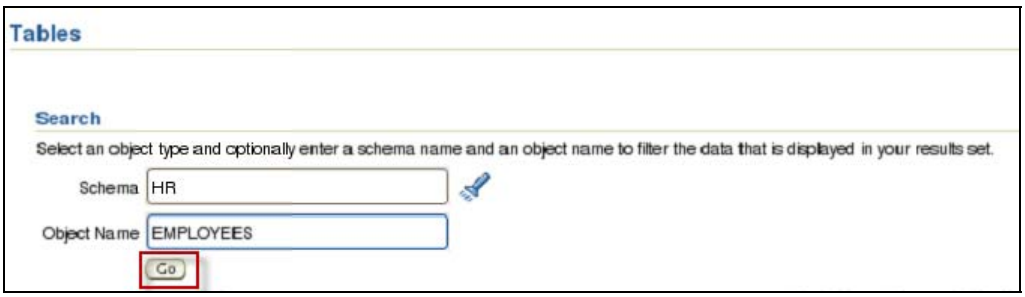
32. Navigate EM and select Targets -> Databases -> db04.oracle.com.



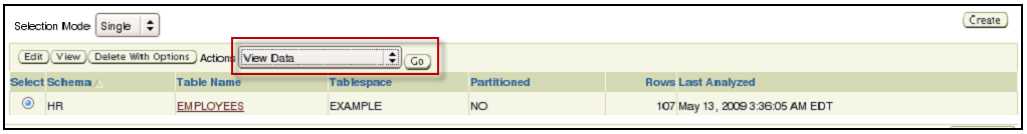
33. Click on the Schema tab for db04.oracle.com and click on Tables under Database Objects.



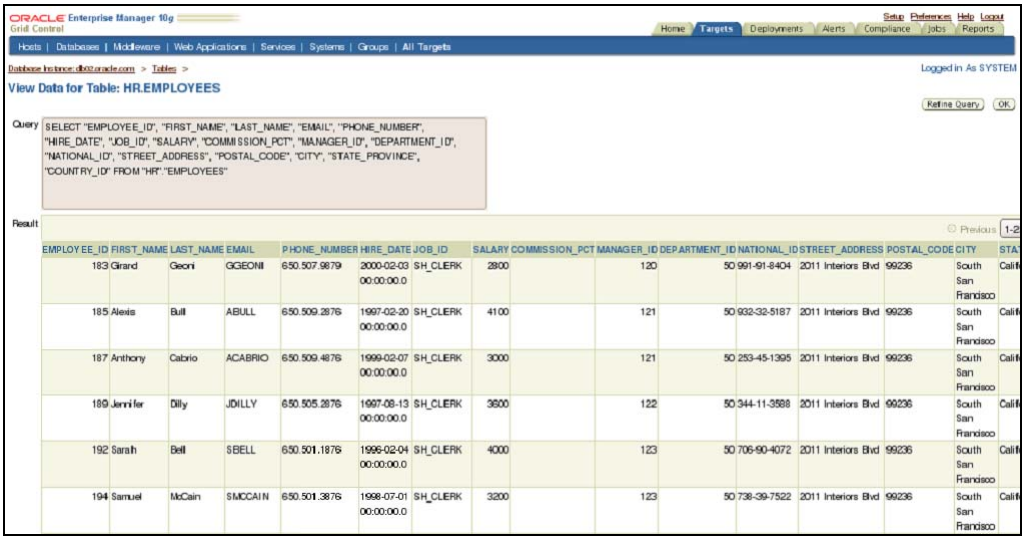
34. If you are brought to the Database Login screen, login as system/oracle1. Leave "Connect As" set to Normal, and then clock the Login button.
35. For the table search, enter HR for the Schema and EMPLOYEES for the Object name.



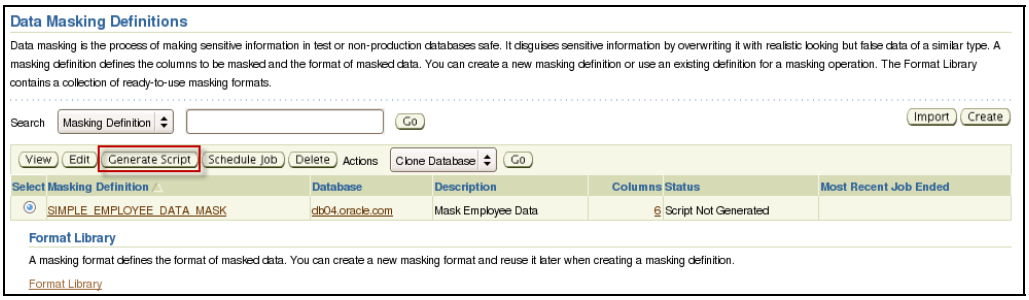
36. Select View Data from the drop-down list box and click on the GO button.



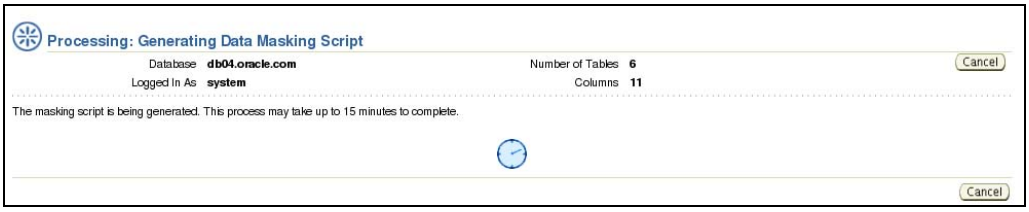
37. Leave this tab open so you can later reference the data before the data masking operation is executed.



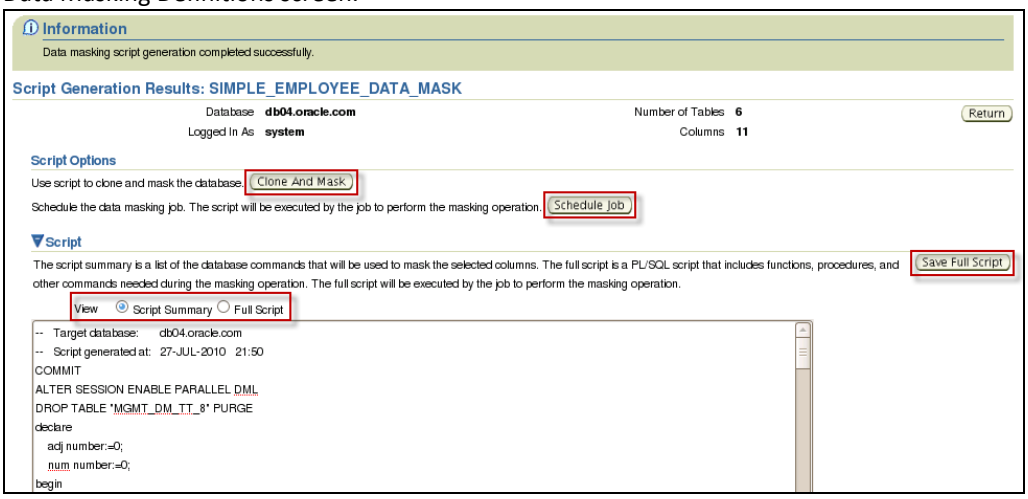
38. Navigate back to the first browser tab. The next step is to select the SIMPLE_EMPLOYEE_DATA_MASK and click on the Generate Script button.



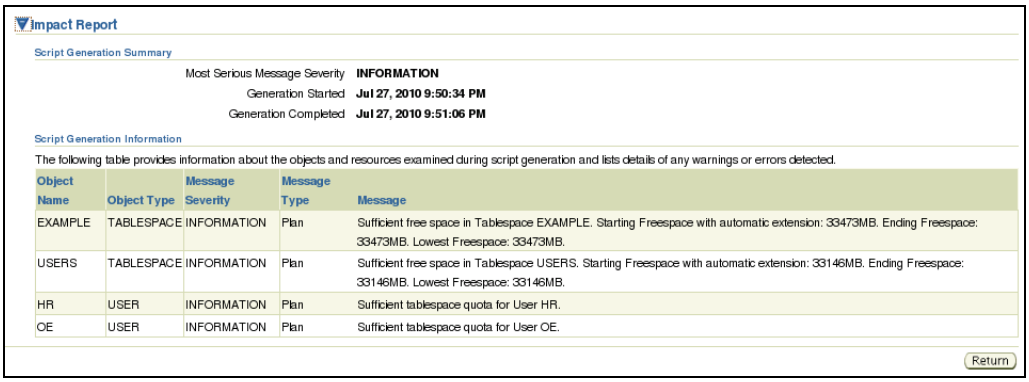
39. After clicking on the Generate Script button, the data masking script will be generated.



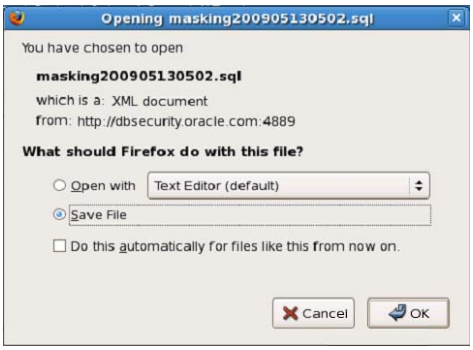
40. You will be forwarded to the Script Generation Results page. There are a number of areas to explore. All of the highlighted buttons and actions can also be accessed on the Data Masking Definitions screen.



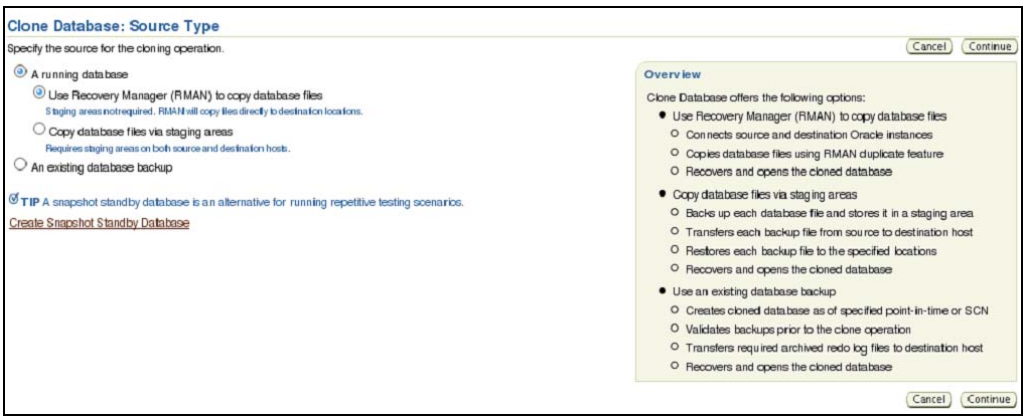
41. Scroll down to the bottom of the page and expand the Impact Report section. The Impact Report will provide a summary of the script generation and important details about the objects and resources necessary to complete the job successfully. If there are any issues here, they should be corrected before moving forward.



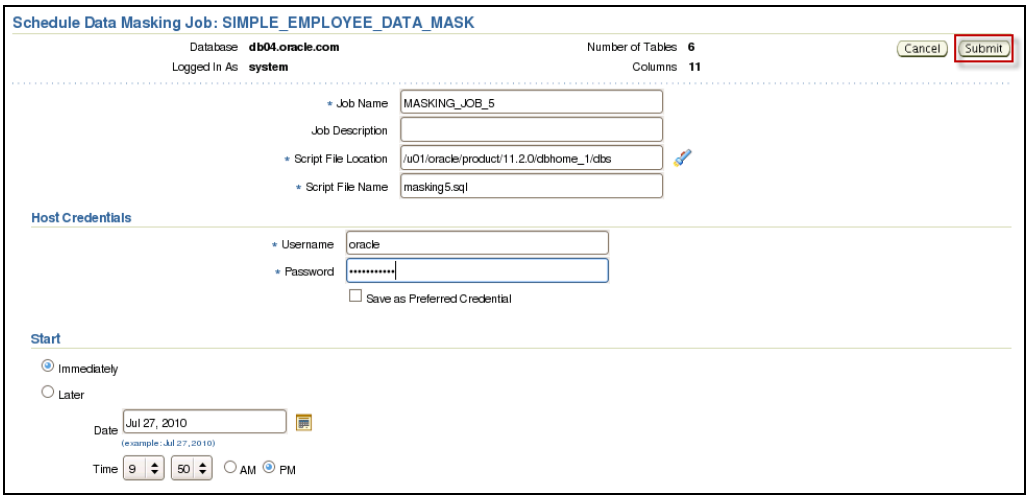
42. Scroll back up the page and click on the Save Full Script button. Take note of the file name of the .sql file to review in detail later. This script could be taken and executed on other targets.



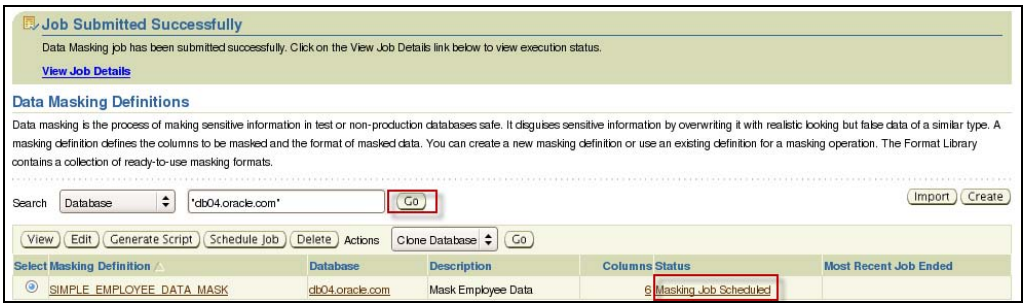
43. Click on the Clone Mask button under the Script Option section. Review the number of supported options to clone the database and create a staging environment for the script to be executed and data to be masked.



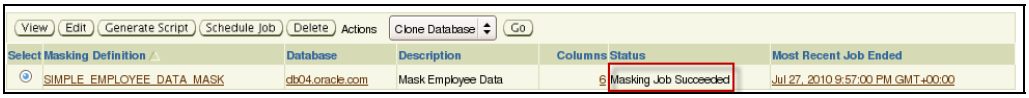
44. Click on the browser's back button to return to the previous screen and click on the Schedule Job button to immediately schedule and run the masking operation. Provide the Host Credentials using the user: Oracle and the provided password. Click on the Submit button to execute the job.



45. Once you submit the job, you will be forwarded to a confirmation page that the job was submitted successfully. Click on the GO button to refresh the status of the job.



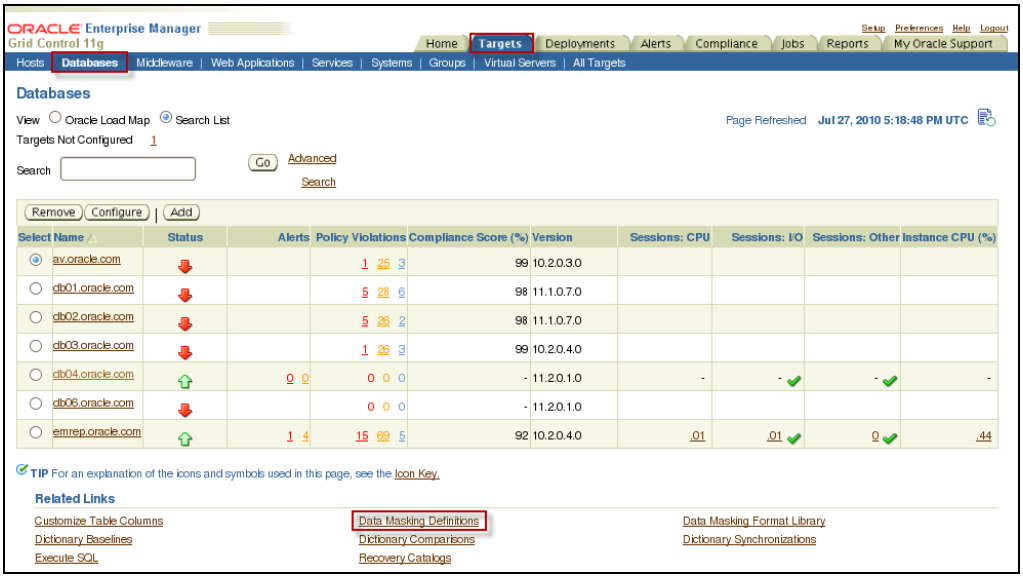
46. Once the job successfully completes, Repeat step 30 to 36 to create a new tab and query the masked data for a before and after comparison.



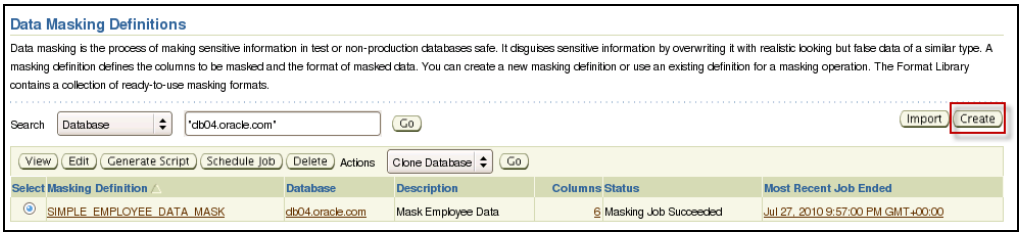
47. Toggle between the two browser tabs and review the data before the masking job and after the successful masking operation of the 5 columns defined.

Using compound masking, condition-based masking and user defined masking

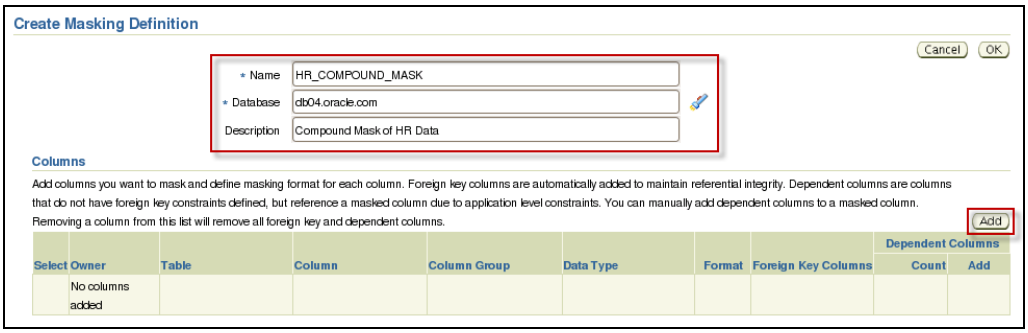
1. Navigate to the Data Masking Definitions by selecting **Targets -> Databases -> Data Masking Definitions**.



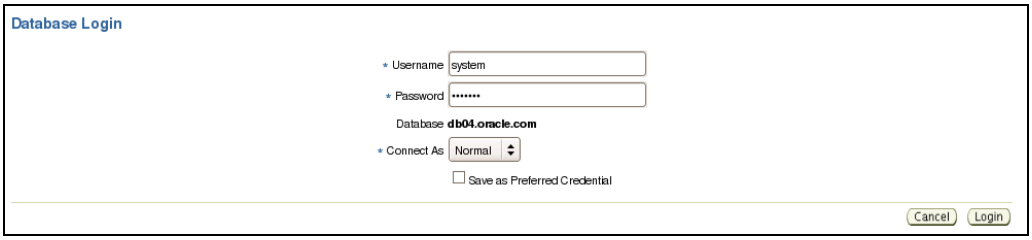
2. From the **Data Masking Definitions** Dialog, we will create a new definition to create a Compound Mask with the **HR.EMPLOYEES** table. Click on the **Create** button to begin the process of creating a new data mask.



3. From the **Create Masking Definition** screen, type in the **Name**, **Database** and **Description** field with the provided values below. Continue and click on the **Add** button.
   i. **Name:** HR_COMPOUND_MASK
      Database: db04.oracle.com
      Description: Compound Mask of HR Data

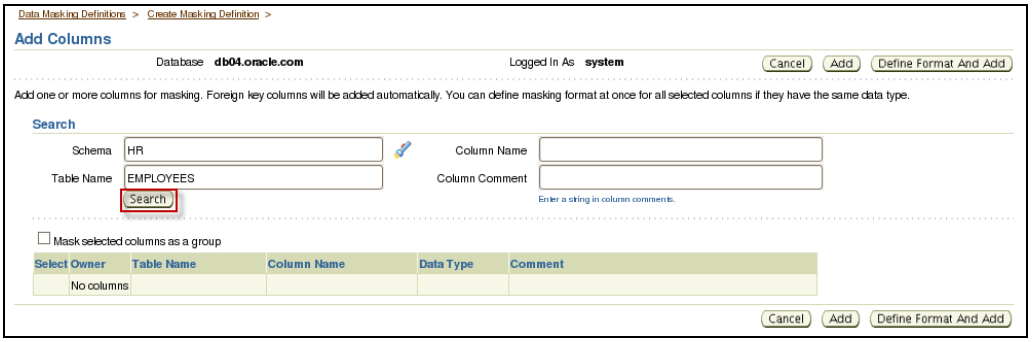


4. If you are brought to the **Database Login** screen, login as **system/oracle1**. Leave "Connect As" set to Normal, and then clock the **Login** button.

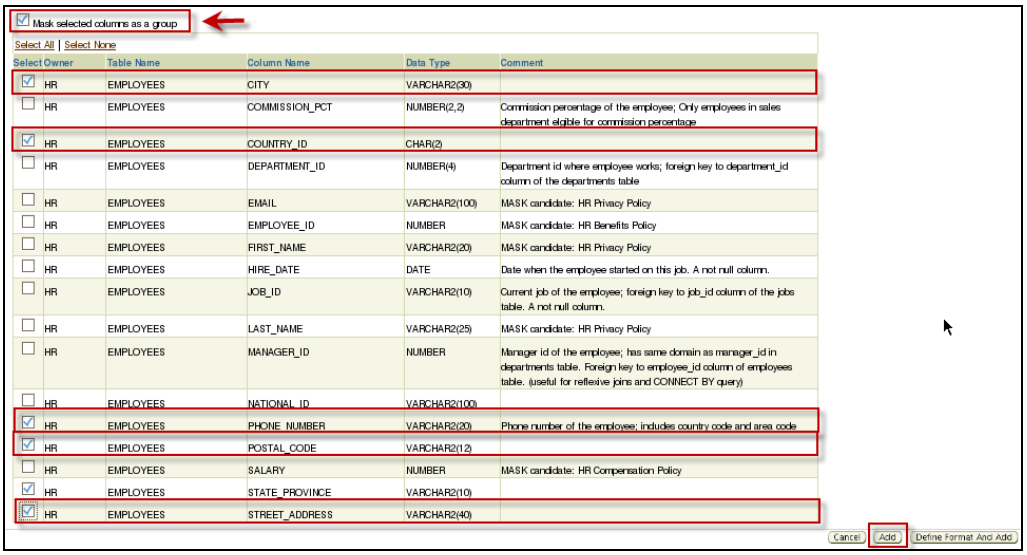


5. We are going to search for the **EMPLOYEES** table in the HR Schema. Type in the following values and click on the **Search** button.

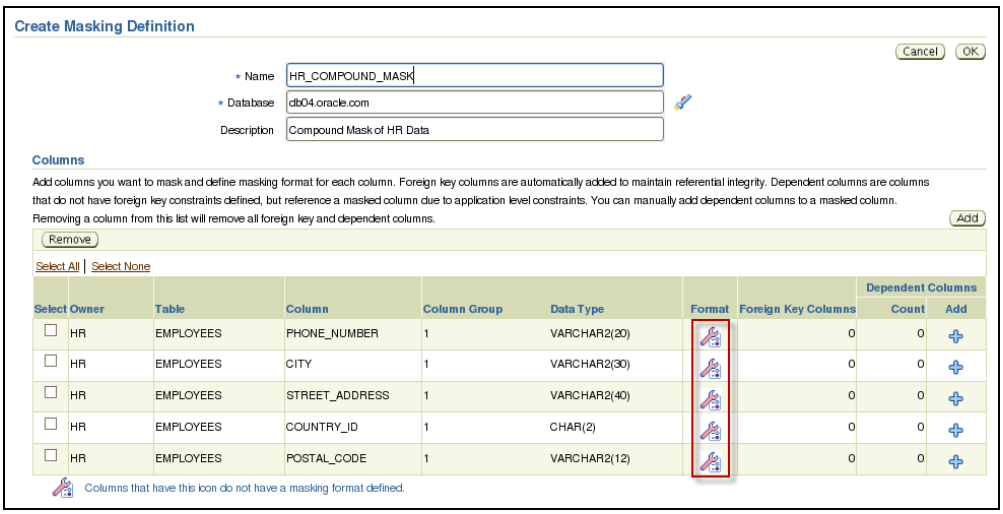
   Schema: HR
   Table Name: EMPLOYEE



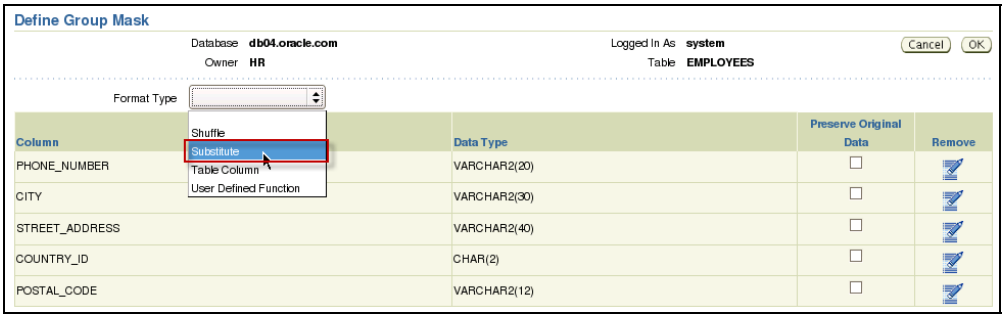
6. Select the columns to be included in the mask. They are: **CITY, COUNTRY_ID, PHONE_NUMBER, POSTAL_CODE, STATE_PROVINCE, STREET_ADDRESS**. Check the box to **"Mask selected columns as a group"** to specify that you want to use mask these columns as a compound mask and continue by clicking on the **Add** button.



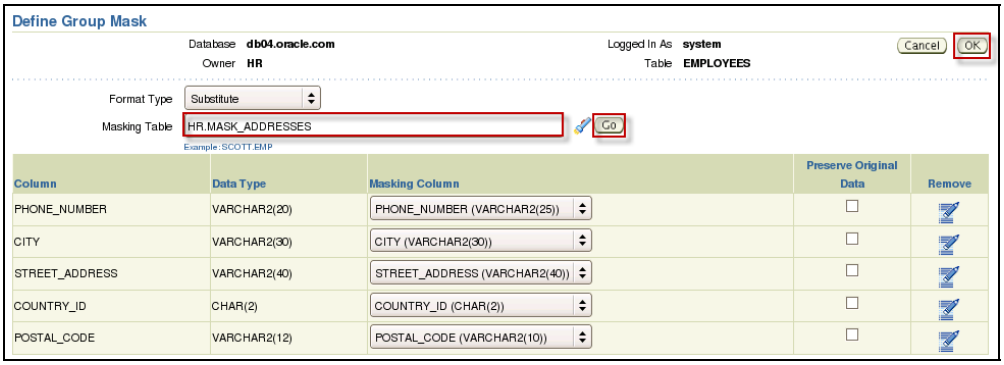
7. Continue by clicking on any of the  Format icons.



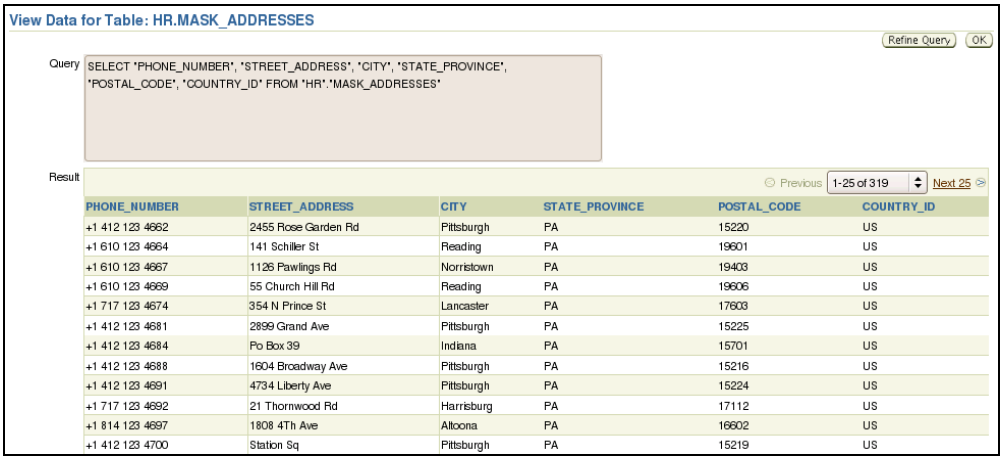
8. In the Define Group Mask screen, select the Format Type **Substitute**. By selecting Substitute, you are defining a deterministic mask—allowing a consistent masking across databases for these columns selected.



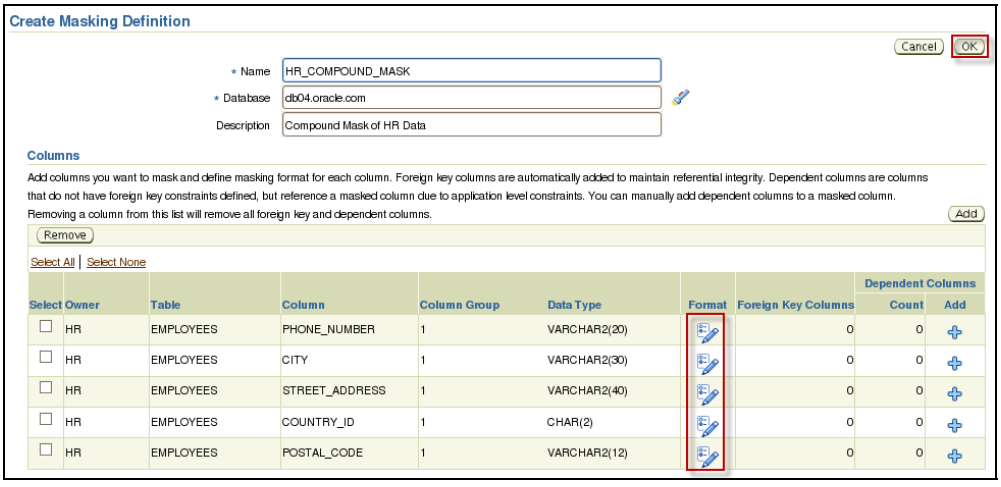
9. Type **HR.MASK_ADDRESSES** in the Masking Table and click on the **Go** button. Select the corresponding Masking Columns from the drop-down list boxes and click on the **OK** button to continue.



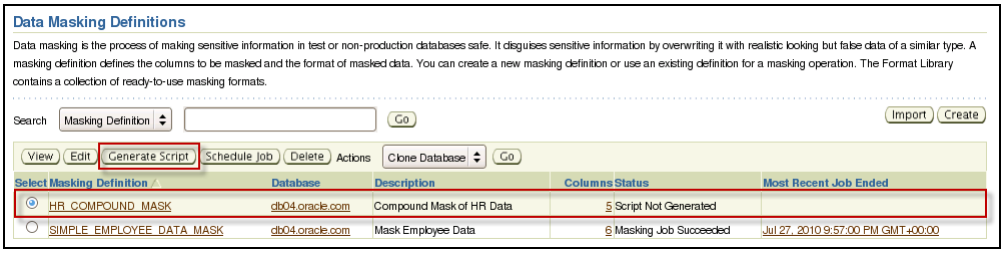
10. As an option, open another tab on your browser window and view the full contents of the **HR.MASK_ADDRESSES** table. We have provided a screen shot so you can see a sample of the data.

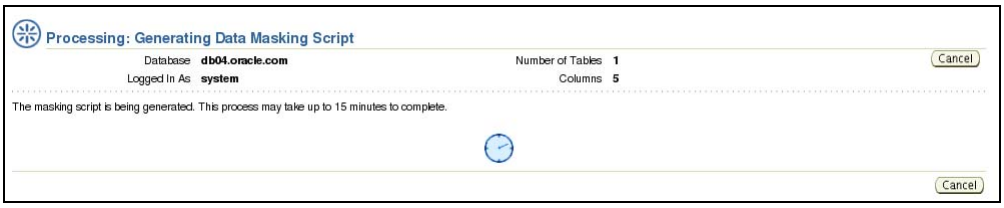


11. Notice that all of the Formats have all been defined. At this step, you could continue to add to your Masking Definition. To finish creating a Masking Definition, click the **OK** button.

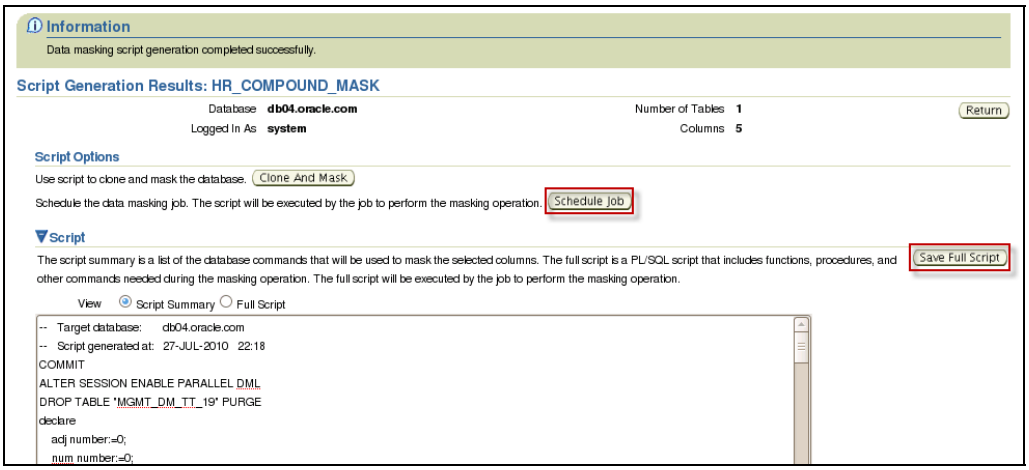


12. You will be brought back to the Data Masking Definitions page. Select the **HR_COMPOUND_MASK** and click on the **Generate Script** button.



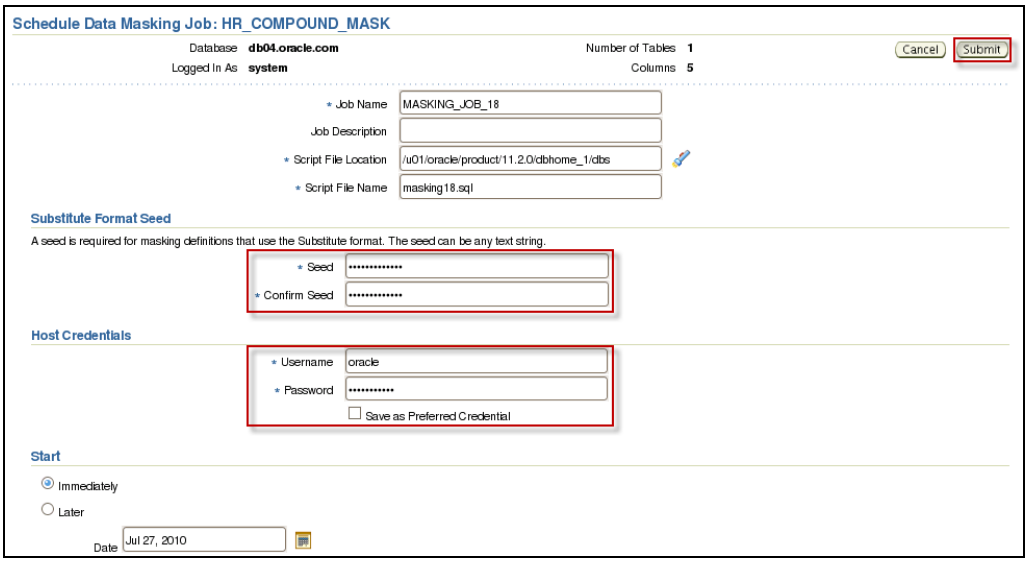


13. After the data masking script generation has completed successfully, scroll down the page and expand the **Impact Report** section. Choose to save the script to disk for additional review by clicking on the **Save Full Script** button.

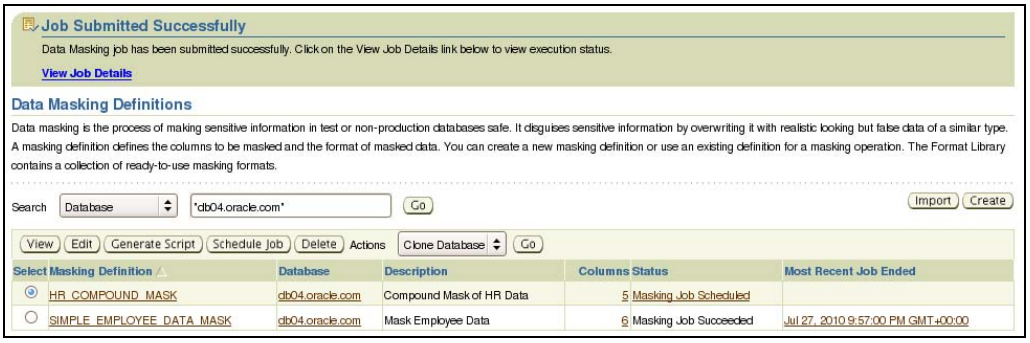


14. Before executing the newly created compound data masking script as we have done previously, open up another browser tab to query the before state of the **HR.EMPLOYEES** table we will be masking.

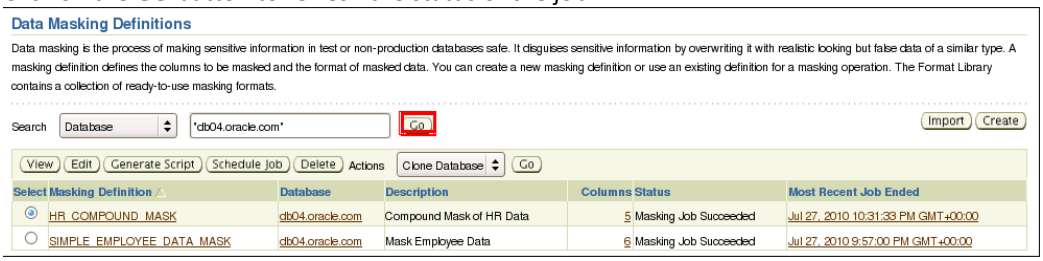
15. Click on the **Schedule Job** button to execute the newly created data mask immediately schedule and run the masking operation. Provide and confirm a Substitute Format Seed, for example, string "**123456**". Provide the Host Credentials using the user: Oracle and the provided password. Click on the **Submit** button to execute the job.



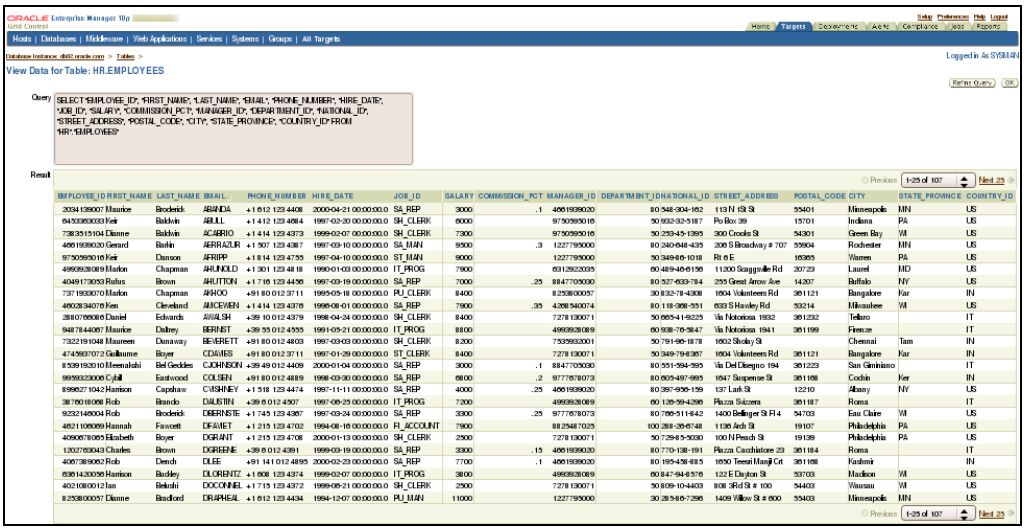
16. Once you submit the job, you will be forwarded to a confirmation page that the job was submitted successfully.



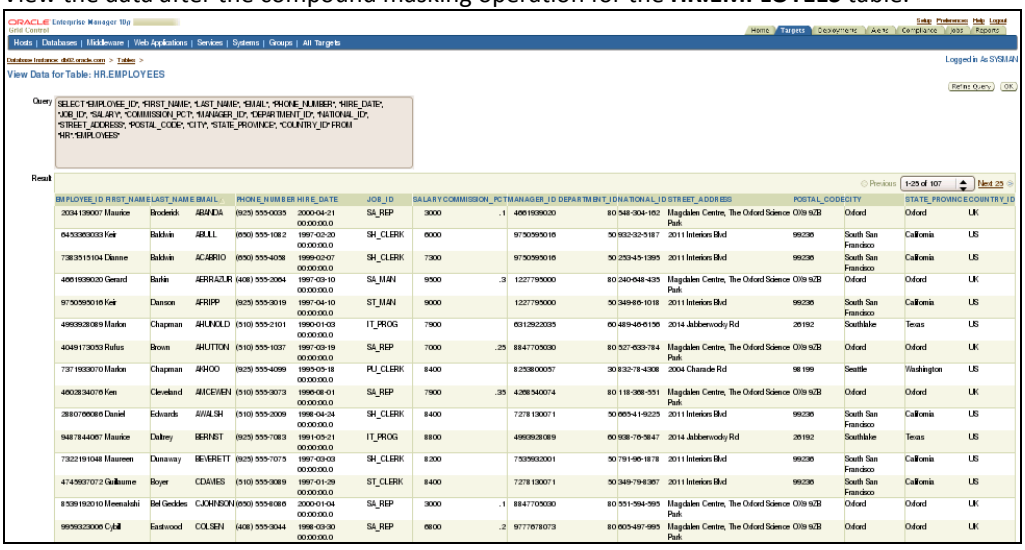
17. Click on the **GO** button to refresh the status of the job.



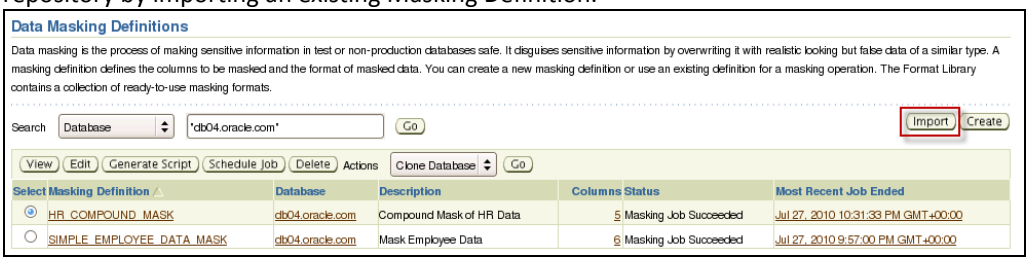
18. Once the job successfully completes, follow the provided steps again to create a new tab and query the masked data for a before and after comparison. View the data before the compound masking operation for the **HR.EMPLOYEES** table.



19. View the data after the compound masking operation for the **HR.EMPLOYEES** table.

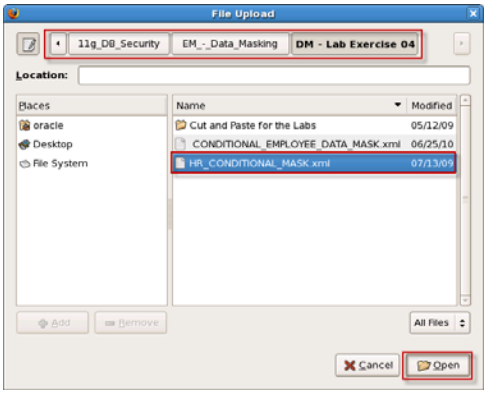


20. Return to the **Data Masking Definition** screen. We will create a new definition in this repository by importing an existing Masking Definition.

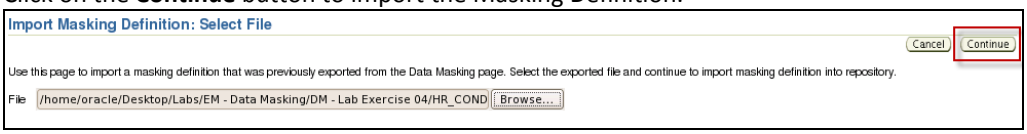


21. Click on the **Browse** button to select the Masking Definition.

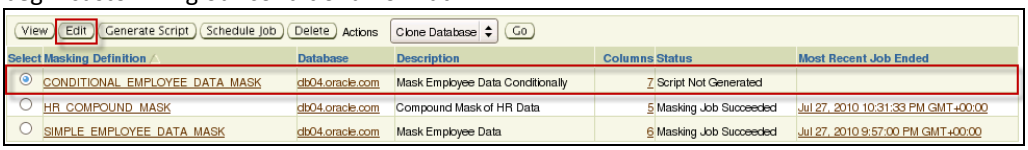
22. Navigate to the folder **oracle->Desktop->Labs->11g_DB_Security->EM – Data_Masking ->DM – Lab Exercise 04**, and select the file named **HR_CONDITIONAL_MASK.xml** Click on the **Open** button to continue.



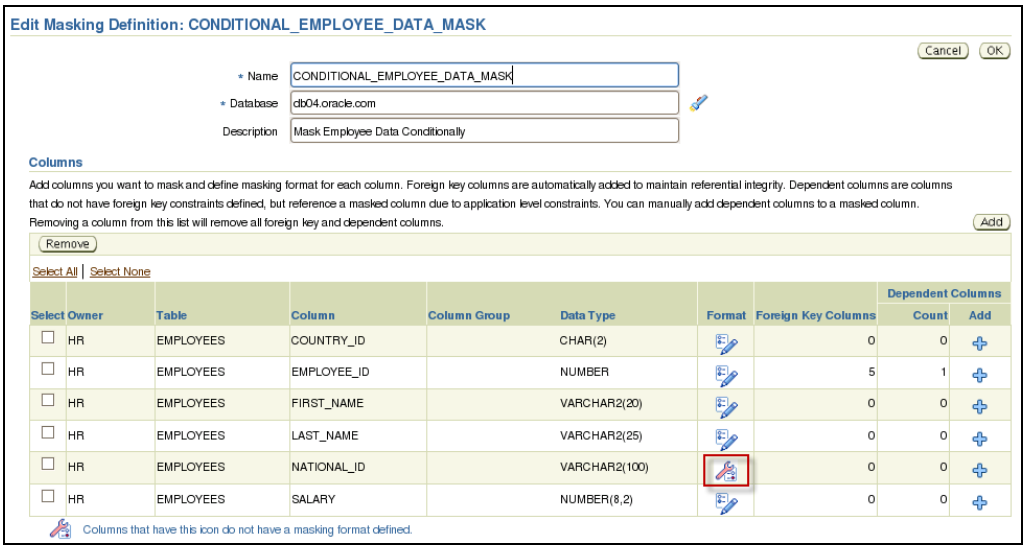
23. Click on the **Continue** button to import the Masking Definition.



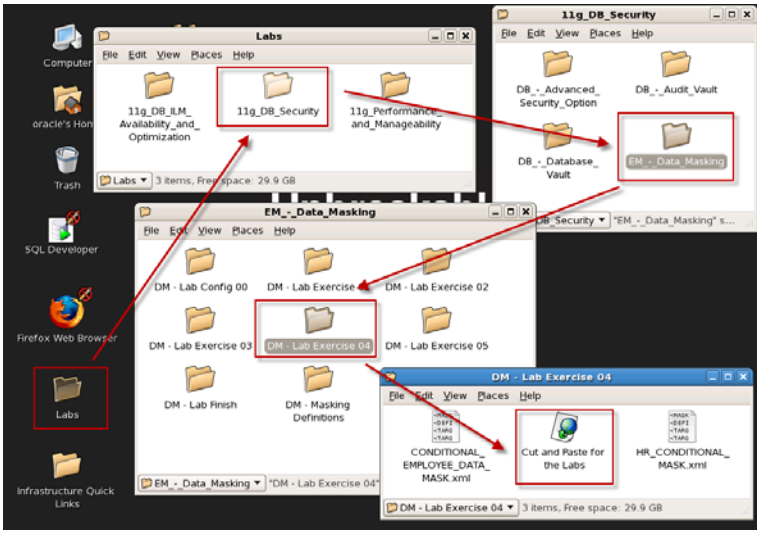
24. With **CONDITIONAL_EMPLOYEE_DATA_MASK** selected, click on the **Edit** button to begin customizing our conditional format.



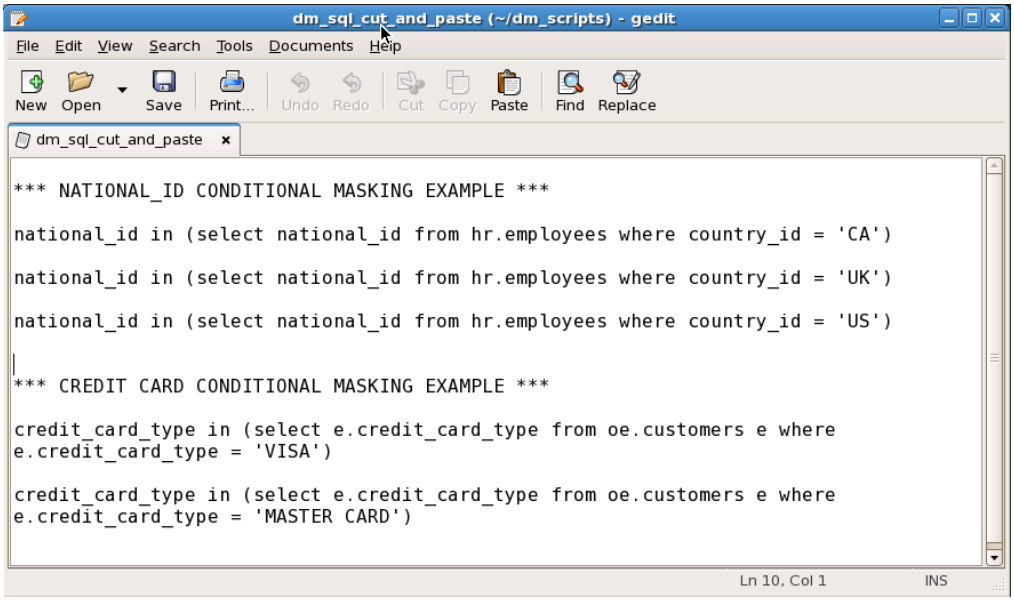
25. For this conditional mask, we want to mask the **NATIONAL_ID** column based upon the **COUNTRY_ID** column. To configure the **NATIONAL_ID** column format, click on the  Format icon.



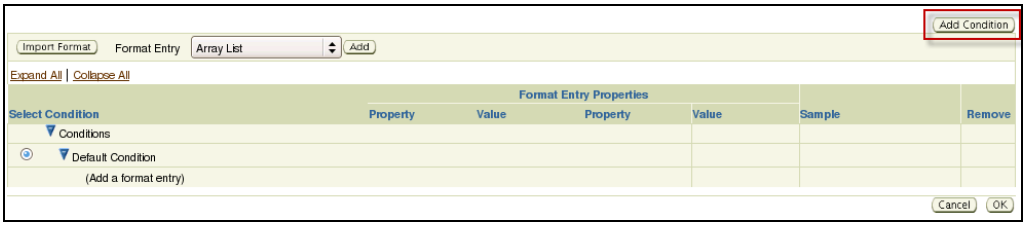
26. To avoid typing errors, navigate the lab folders to access a text document with the correct SQL conditional text. Navigate the folders **oracle->Desktop->Labs->11g_DB_Security->EM – Data_Masking ->DM – Lab Exercise 04**.



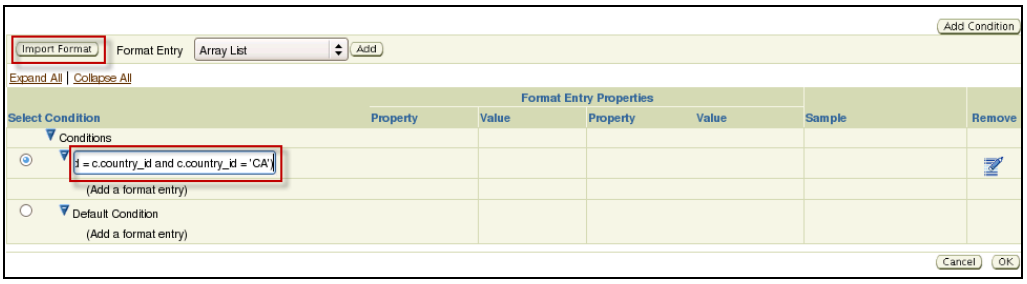
27. Double click on the icon **Cut and Paste for the Labs** document and open up in the emacs editor. This SQL will be used to evaluate our conditions for proper masking.



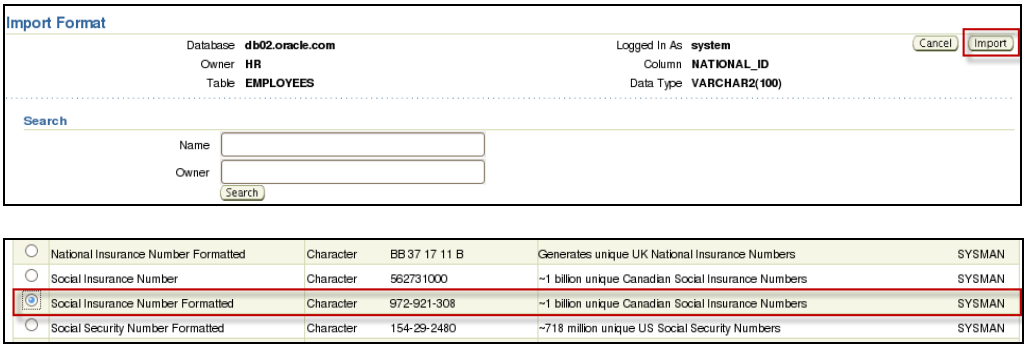
28. In the **Define Column Mask** screen, click on the **Add Condition** button. We will be adding 3 conditions based upon the SQL to test for the **COUNTRY_ID** value.



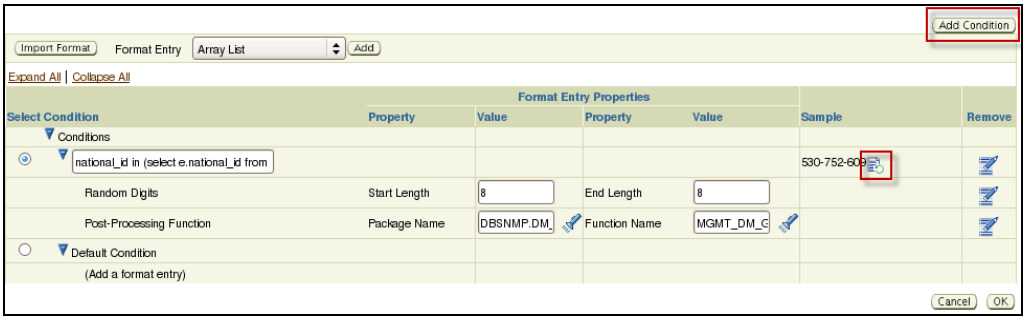
29. Type (cut and paste) the following SQL Condition and click on the **Import Format** button.
    1) national_id in (select national_id from hr.employees where country_id = 'CA')



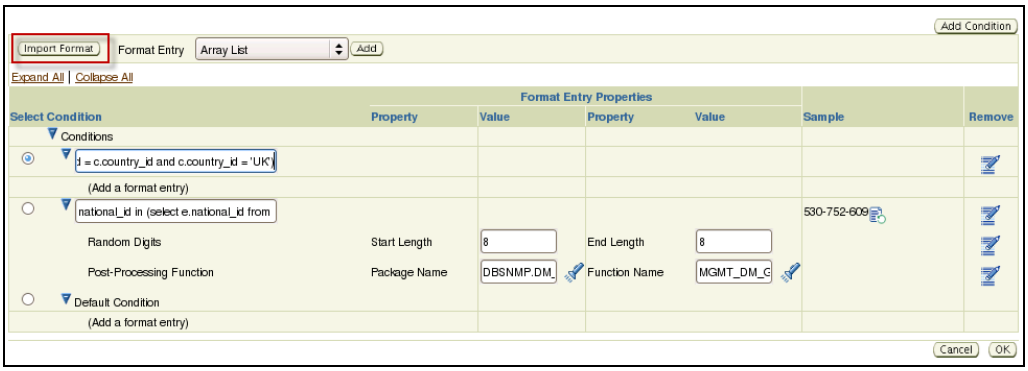
30. If the condition is met that the **COUNTRY_ID** value is '**CA'**, then we will use the Canadian **Social Insurance Number Formatted** provided out of the box with the product. Select the corresponding radio button and click on the **Import** button.



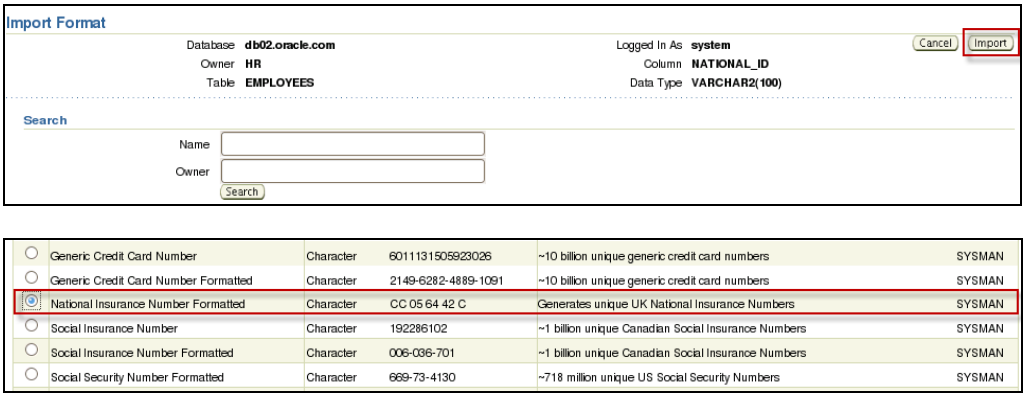
31. Review the Masking Format. Click on the Sample icon to view sample data and continue by clicking the **Add Condition** button.



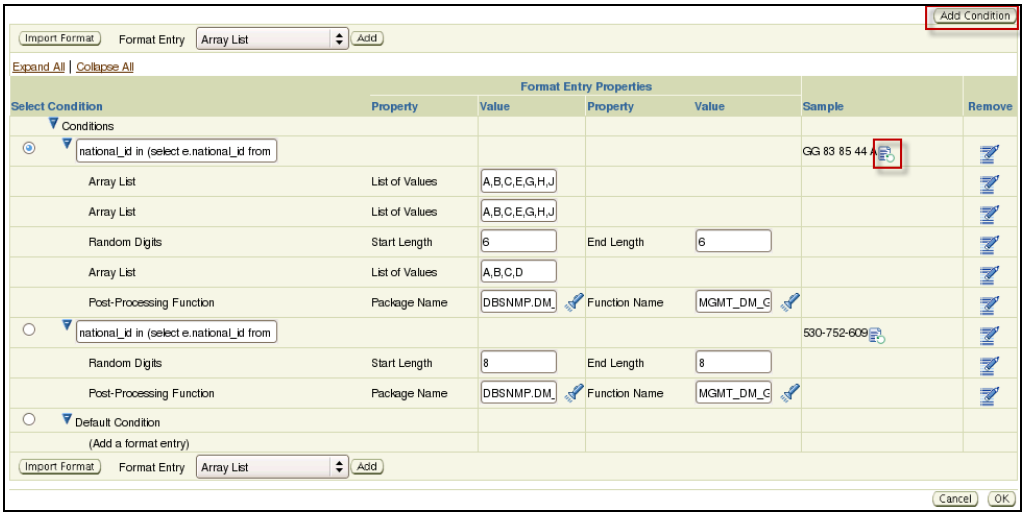
32. Add the second Conditional Masking definition. Type (cut and paste) the following SQL Condition and click on the **Import Format** button.
    1) national_id in (select national_id from hr.employees where country_id = 'UK')



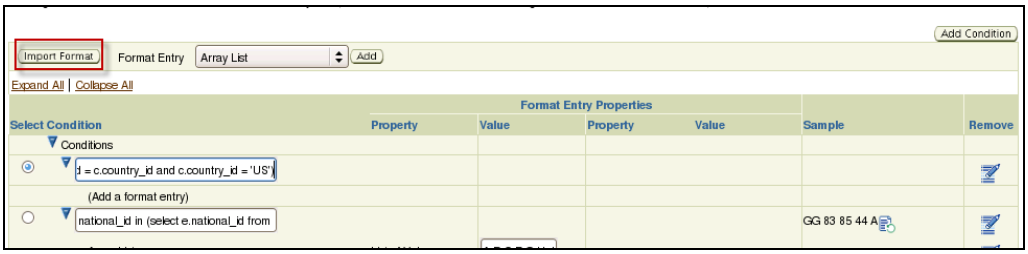
33. If the condition is met that the **COUNTRY_ID** value is '**UK'**, then we will use the **National Insurance Number Formatted** provided out of the box with the product. Select the corresponding radio button and click on the **Import** button.



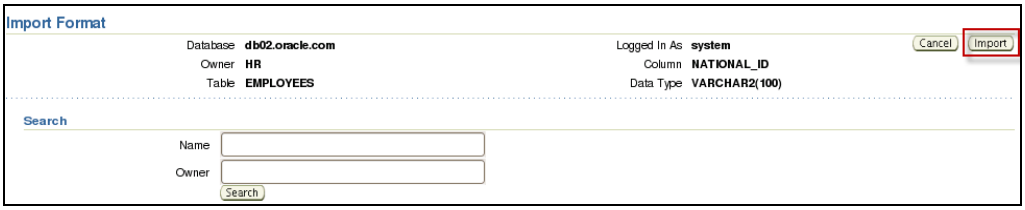
34. Review the Masking Format. Click on the Sample icon  to view sample data and continue by clicking the **Add Condition** button.

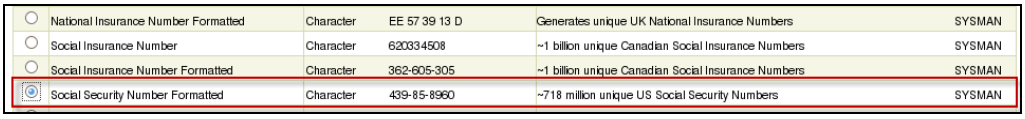


35. Add the third Conditional Masking definition. Type (cut and paste) the following SQL Condition and click on the **Import Format** button.
    1) national_id in (select national_id from hr.employees where country_id = 'US')

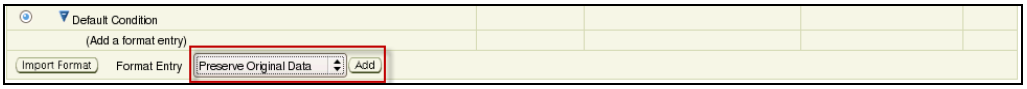


36. If the condition is met that the **COUNTRY_ID** value is '**US'**, then we will use the **Social Security Number Formatted** provided out of the box with the product. Select the corresponding radio button and click on the **Import** button.

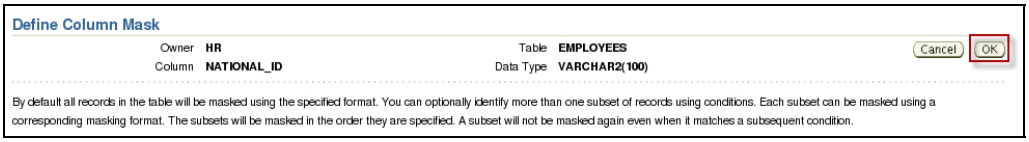




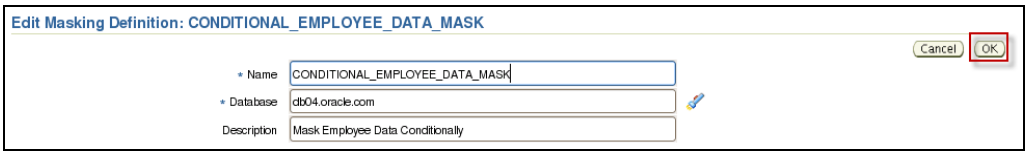
37. The last step is to set the Default mask if the value of **COUNTRY_ID** is not met by any of our conditions, either '**CA', 'UK'** or **'US'**. Select the radio button for the **Default Condition** and choose the **Preserve Original Data** mask and click on the **Add** button.



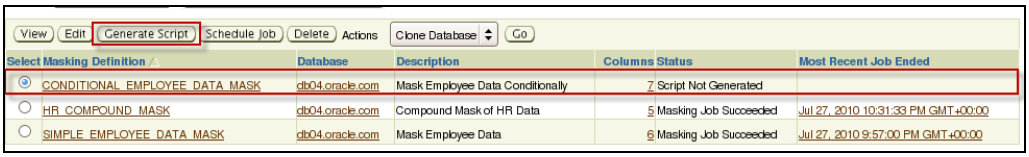
38. To finish defining a Column Mask, click the **OK** button.

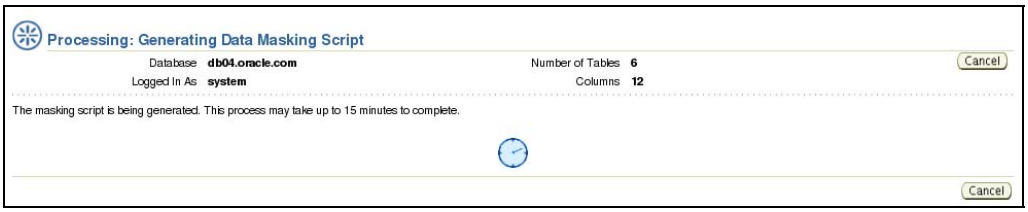


39. The creation of our Condition-based mask is now complete. To finish editing the Masking Definition, click the **OK** button.

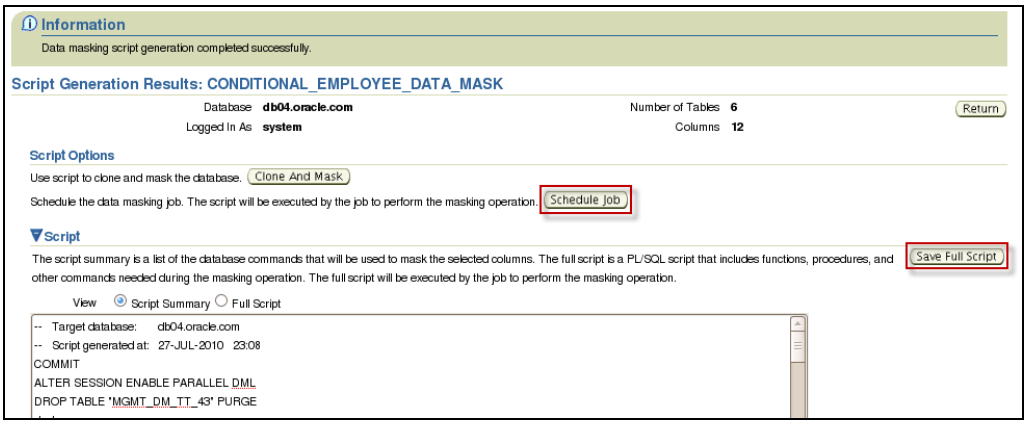


40. You will be brought back to the Data Masking Definitions page. Select the **CONDITIONAL_EMPLOYEE_DATA_MASK** and click on the **Generate Script** button.



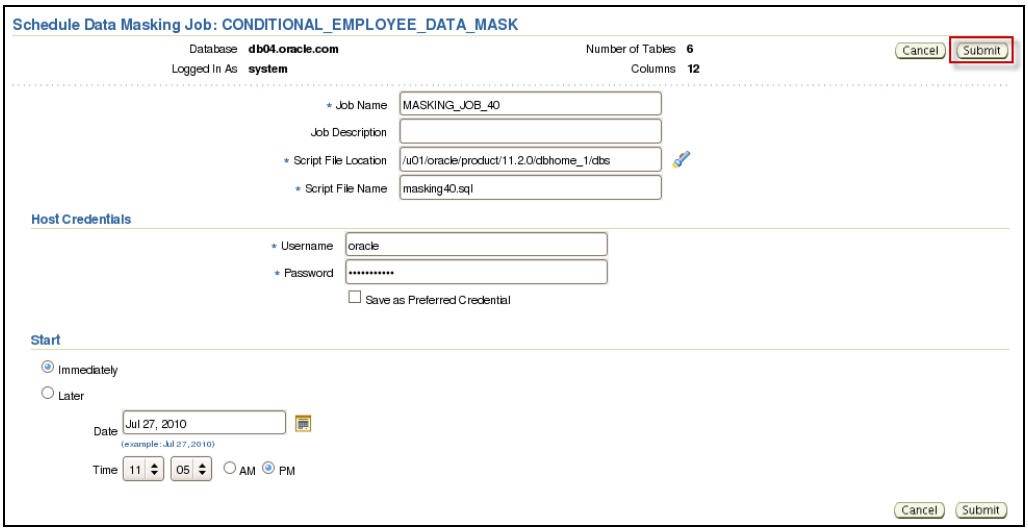


41. After the data masking script generation has completed successfully, scroll down the page and expand the **Impact Report** section. Choose to save the script to disk for additional review by clicking on the **Save Full Script** button.

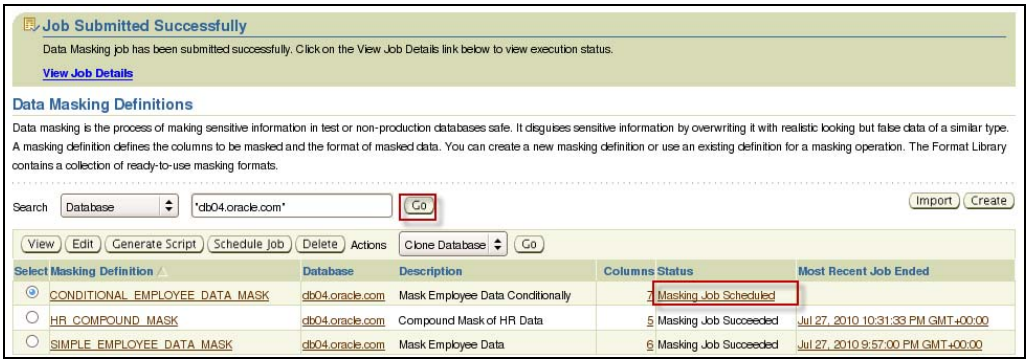


42. Before executing the newly created Conditional-based data masking script as we have done previously, open up another browser tab to query the before state of the **HR.EMPLOYEES** table we will be masking.

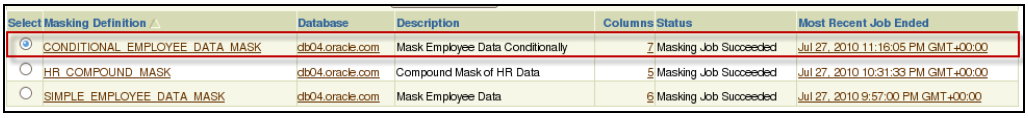
43. Click on the **Schedule Job** button to execute the newly created data mask immediately schedule and run the masking operation. Provide the Host Credentials using the user: Oracle and the provided password. Click on the **Submit** button to execute the job.



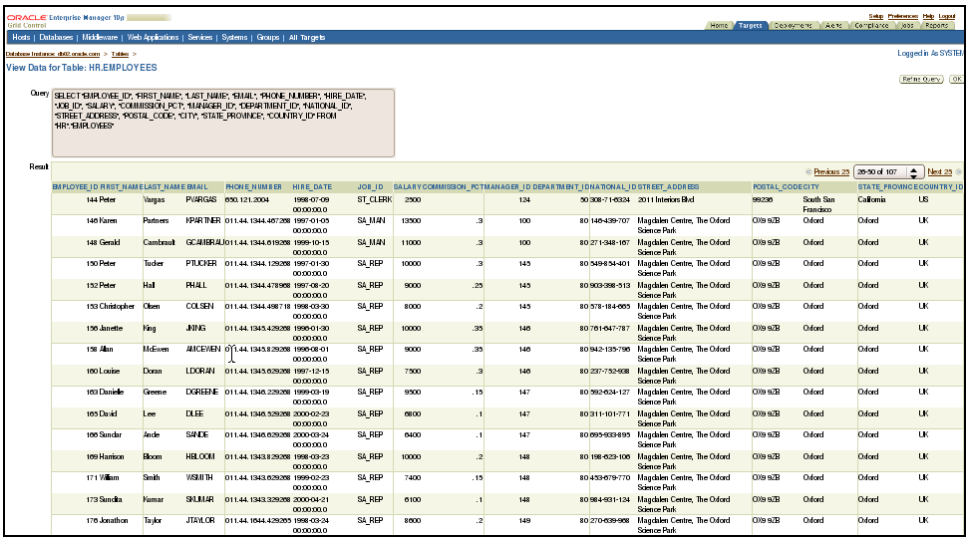
44. Once you submit the job, you will be forwarded to a confirmation page that the job was submitted successfully.



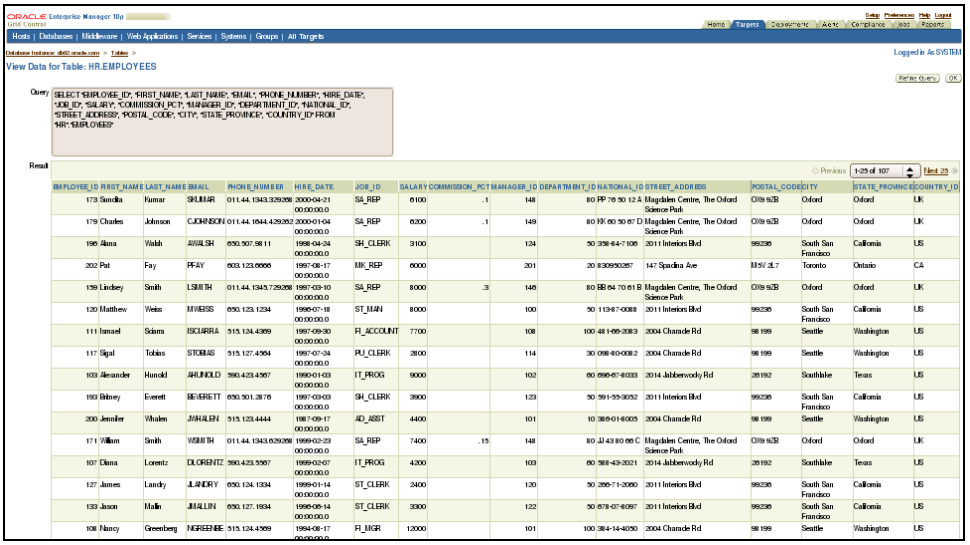
45. Click on the **GO** button to refresh the status of the job.



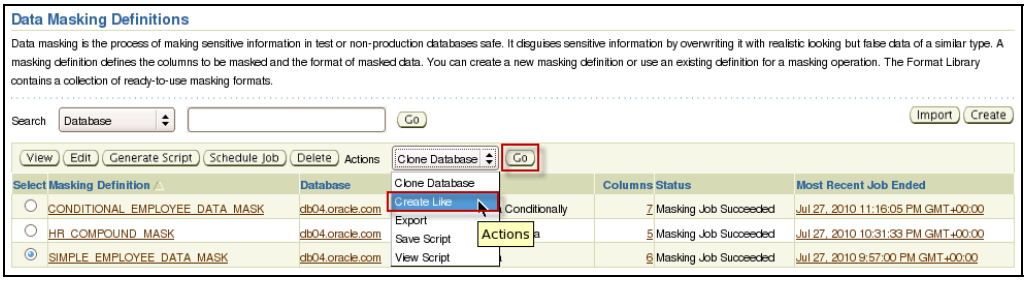
46. Once the job successfully completes, follow the provided steps again to create a new tab and query the masked data for a before and after comparison. View the data before the compound masking operation for the **HR.EMPLOYEES** table.



47. View the data after the compound masking operation for the **HR.EMPLOYEES** table.

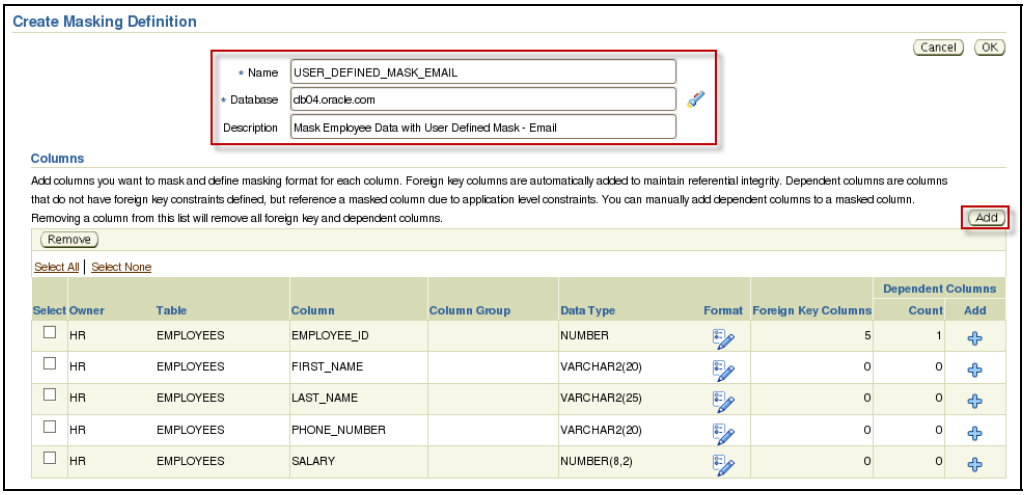


48. Return to the **Data Masking Definition** screen. We will create a new definition by using the Create Like option. Select the Masking Definition **SIMPLE_EMPLOYEE_DATA_MASK** and choose the **Create Like** option from the Actions and click the **Go** button.



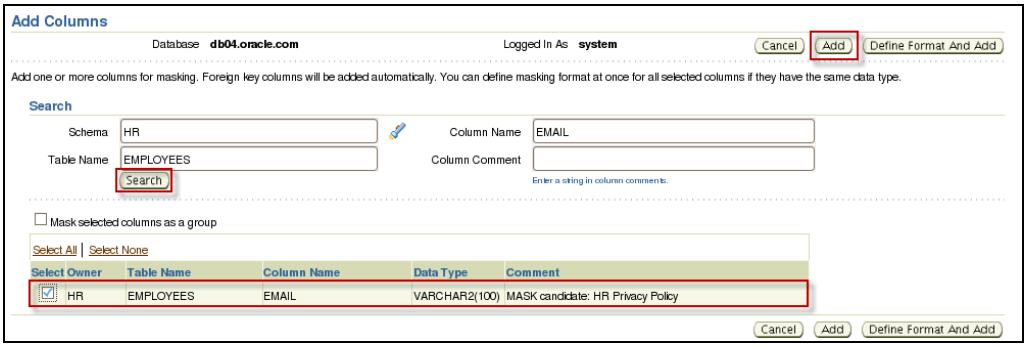
49. From the **Create Masking Definition** screen, type in the **Name**, **Database** and **Description** field with the provided values below. Continue and click on the **Add** button.

i. **Name:** USER_DEFINED_MASK_EMAIL
Database: db04.oracle.com
Description: Mask Employee Data with User Defined Mask - Email

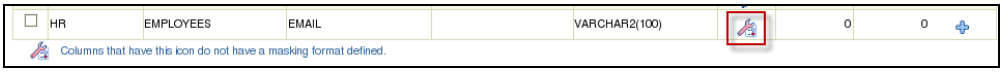


50. For this Masking Definition, we will add the column EMAIL. We are going to search in the EMPLOYEES table in the HR Schema for the EMAIL Column name. Type in the following values and click on the **Search** button. Select the **EMAIL** column and click on the **Add** button.

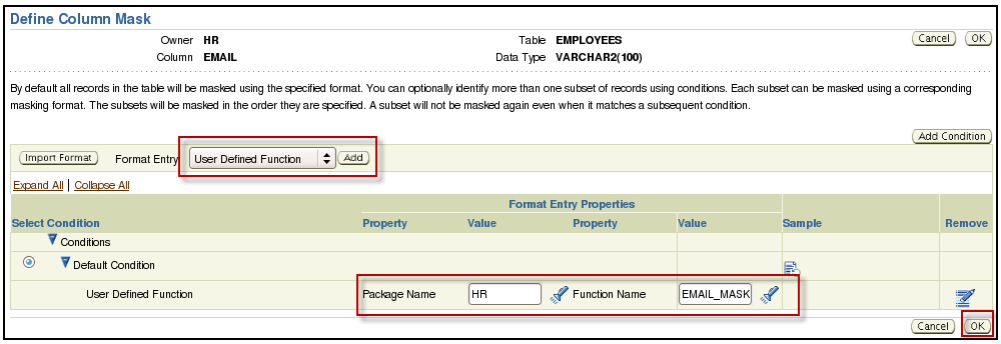
i. **Schema:** HR
Table Name: EMPLOYEE
Column Name: EMAIL



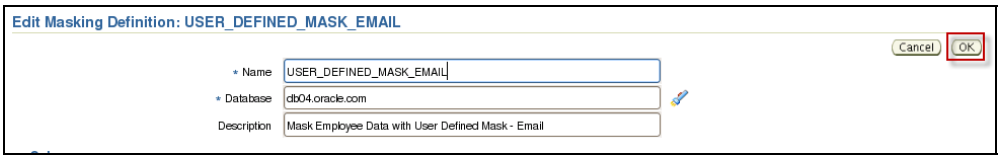
51. Continue by clicking on the  Format icon on the EMAIL column.



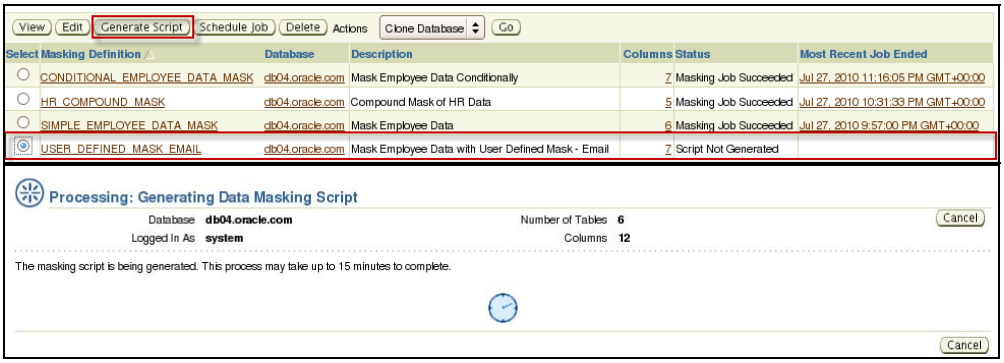
52. In the Define Column Mask screen in the section of Format Entry, select **User Defined Function** from the drop-down list box and click on the Add button. After the **Default Condition** section expands, specify the **Package Name: HR** and **Function Name: EMAIL_MASK**. Click on the **OK** button.



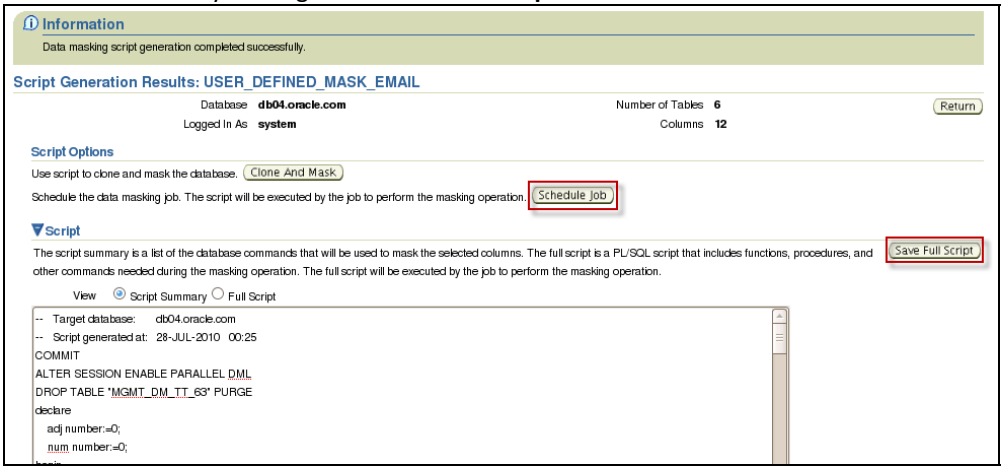
53. At this step, you could continue to add to your Masking Definition. To finish creating this Masking Definition, click the **OK** button.



54. You will be brought back to the Data Masking Definitions page. Select the **USER_DEFINED_MASK_EMAIL** and click on the **Generate Script** button.

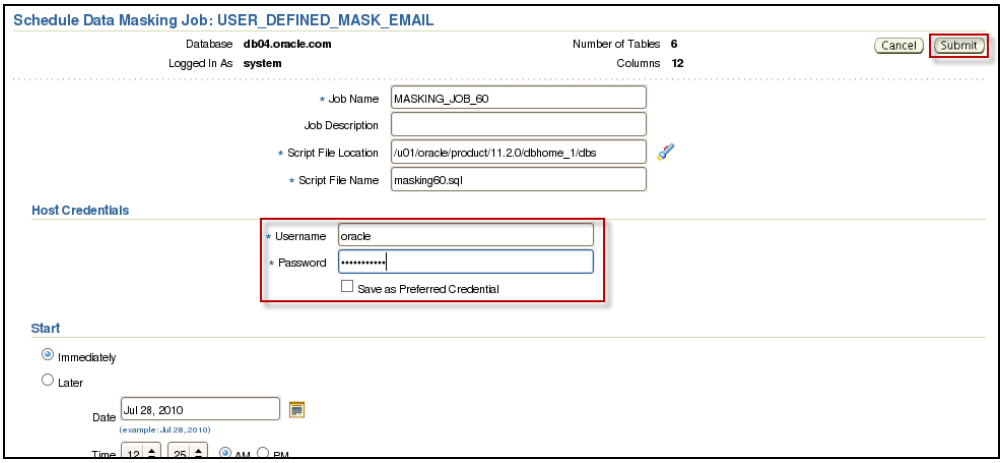


55. After the data masking script generation has completed successfully, scroll down the page and expand the **Impact Report** section. Choose to save the script to disk for additional review by clicking on the **Save Full Script** button.

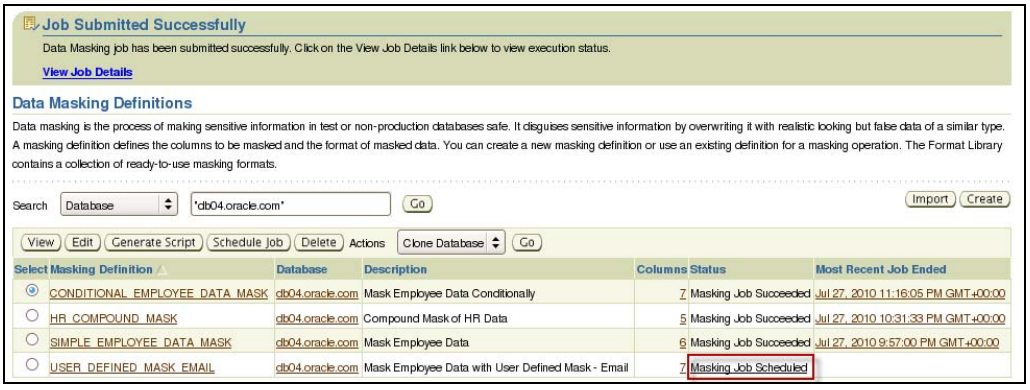


56. Before executing the newly created compound data masking script as we have done previously, open up another browser tab to query the before state of the **HR.EMPLOYEES** table we will be masking.

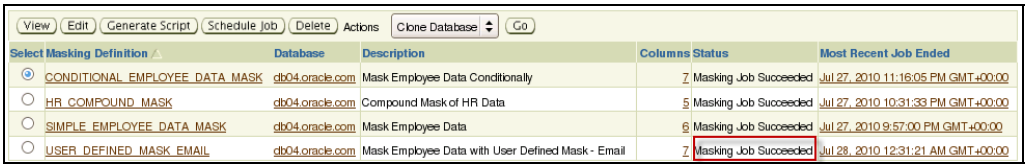
57. Click on the **Schedule Job** button to execute the newly created data mask immediately schedule and run the masking operation. Provide the Host Credentials using the user: Oracle and the provided password. Click on the **Submit** button to execute the job.



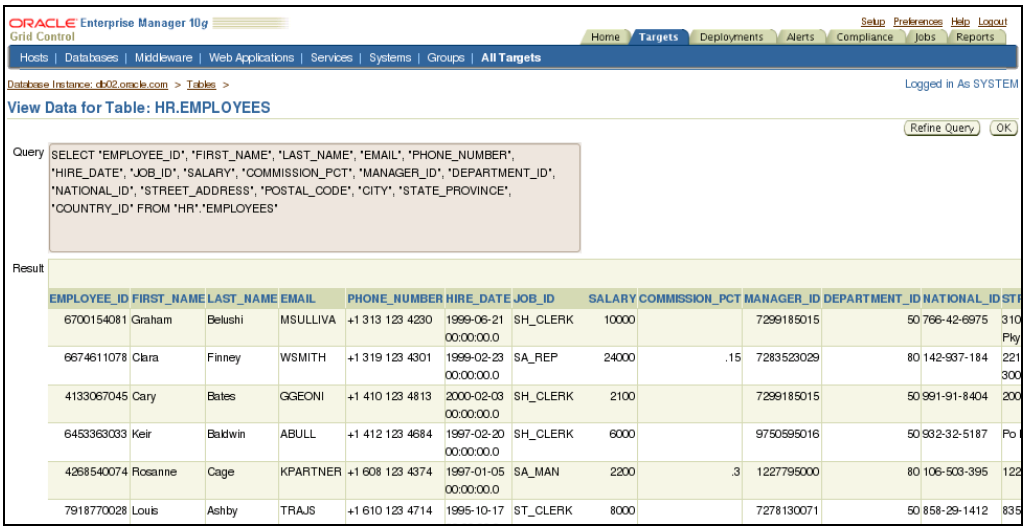
58. Once you submit the job, you will be forwarded to a confirmation page that the job was submitted successfully.



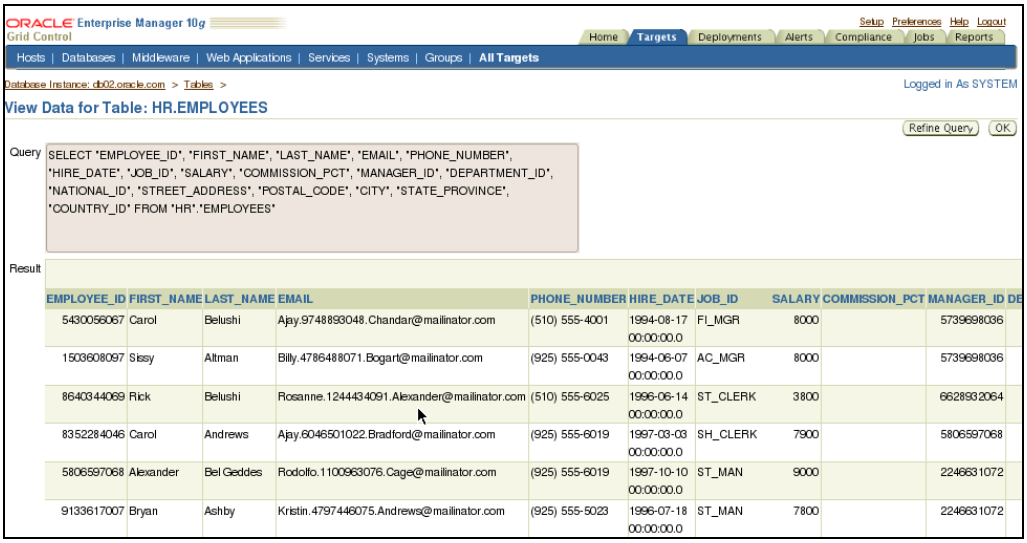
59. Click on the **GO** button to refresh the status of the job.



60. Once the job successfully completes, follow the provided steps again to create a new tab and query the masked data for a before and after comparison. View the data before the user-defined masking operation for the **HR.EMPLOYEES** table on the **EMAIL** column.



61. View the data after the compound masking operation for the **HR.EMPLOYEES** table. Notice the new masked values for **EMAIL** column.

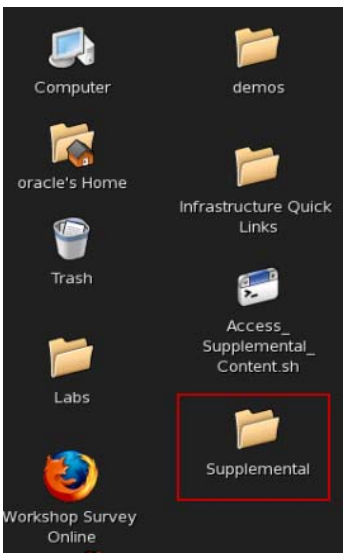


This concludes the Oracle Enterprise Manager Data Masking Hands-on Lab. If you have time, please continue the following OPTIONAL lab for extra credit ☺ !

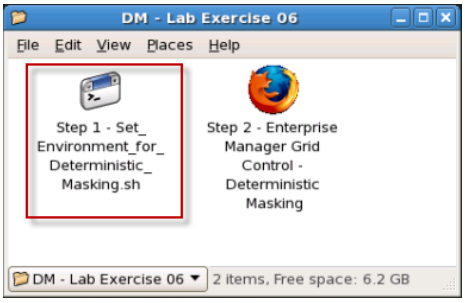
Deterministic masking (OPTIONAL – Extra Credit)

1. Navigate to the desktop and click on the icon, **'Access_Supplemental_Content.sh'**. Click on the **Run in Terminal** button when provided the option.

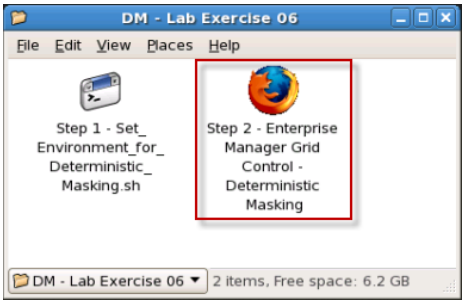
2. After the **Supplemental** folder is copied to the desktop, drill down to the folder **Supplemental->11g_DB_Security → EM_-_Data_Masking → DM – Lab Exercise 06**.



3. In the **DM – Lab Exercise 06** folder, click on the icon, **'Step 1-Set_Environment_for_Deterministic_Masking.sh'** . This will set up the two database users, HR01 and HR02 that will be used in this exercise. Hit the "return" to close the window once the script is done.

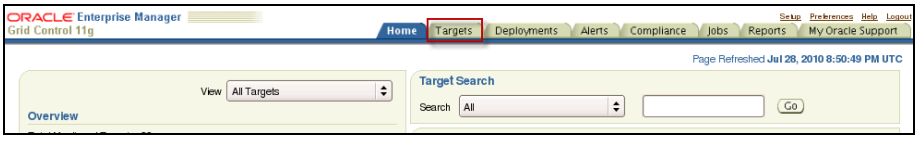


4. Click on **'Step 2 – Enterprise Manager Grid Control – Deterministic Masking'** to open the browser to login into Grid Control.

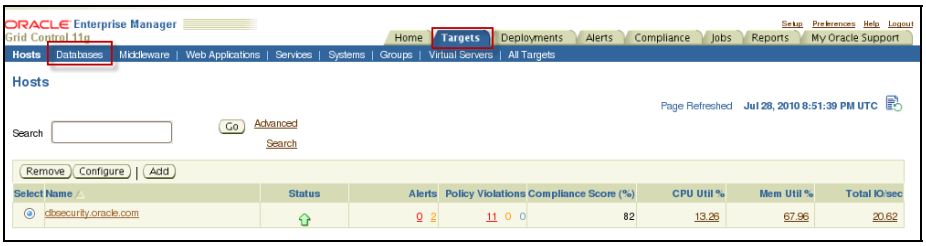


5. Login to Grid Control at the URL http://dbsecurity.oracle.com:4889/em using the User Name: sysman and the Password: oracle1. Click on the Login button.

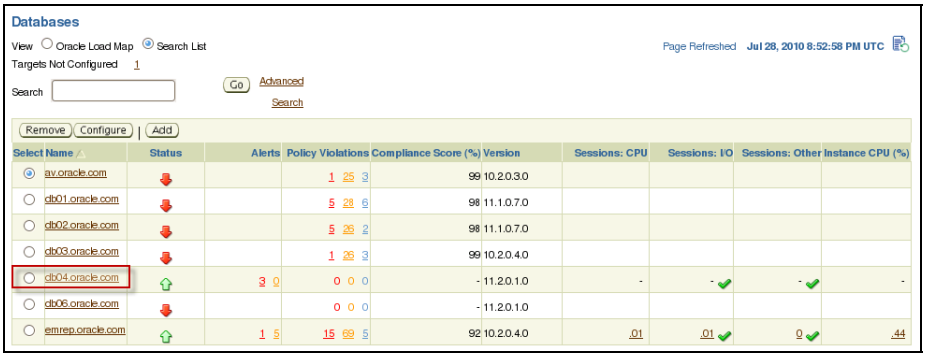
6. After logging on to Enterprise Manager – Grid Control, click on the Targets tab.



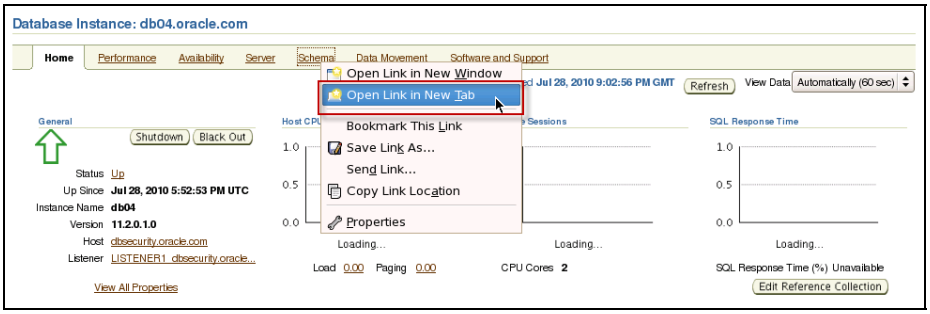
7. Click on Databases.



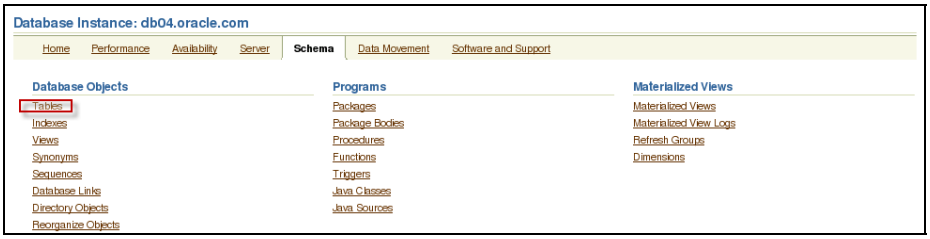
8. Click on the database link db04.oracle.com.



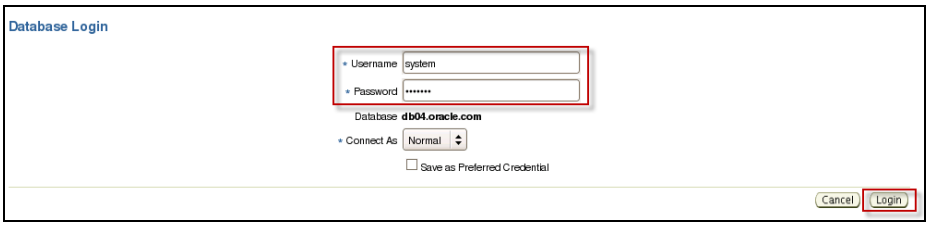
9. Right click on the link, ‘Schema’ and choose the option, ‘Open Link in New Tab’.



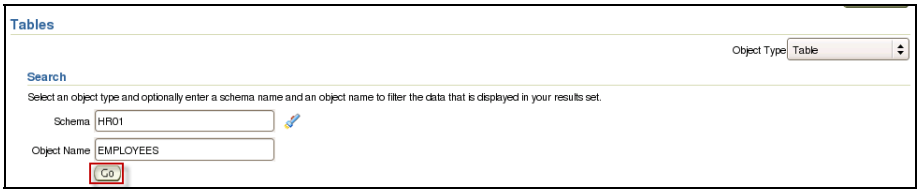
10. In the newly opened tab, we will query the HR01 table. Click on the Tables link.



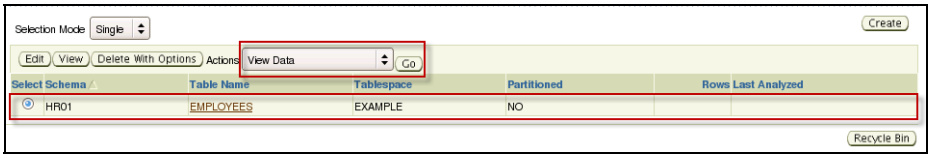
11. If prompted, login to the database using the Username: system and Password: oracle1. Click on the Login button when finished.
TIP: select the option to “Save as Preferred Credential”.



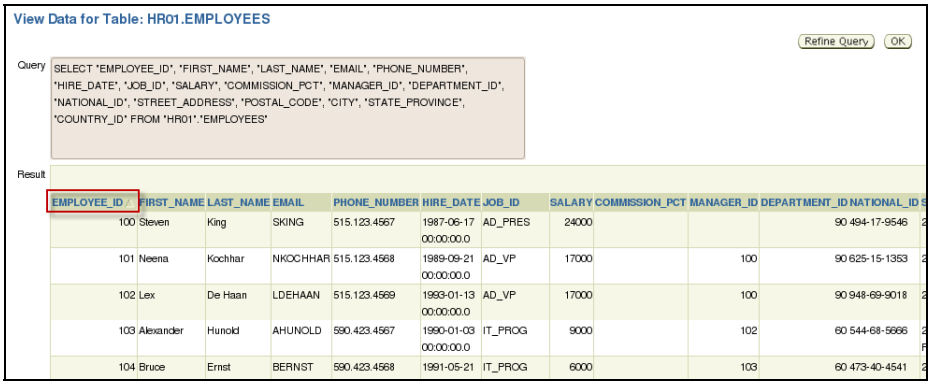
12. We will be querying the HR01 schema and the EMPLOYEES table. Click on the Go button.



13. Select the Action to View Data and click on the Go button.

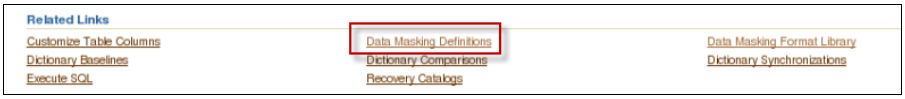


14. Click on the EMPLOYEE_ID column to sort the data. This is the pre-masked data for HR01.EMPLOYEES. We will use the same steps above to view this table after the masking process.

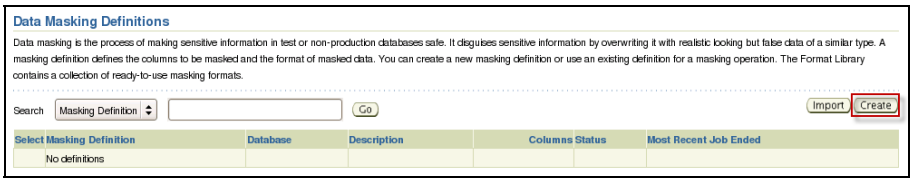


15. Navigate back to the first browser tab. Click on the Databases link.

16. Scroll down to the bottom of the page and select the Data Masking Definitions link.

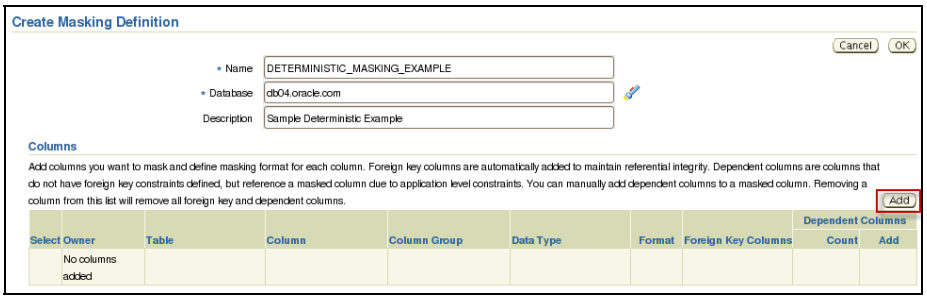


17. In the Data Masking Definitions screen, click on the Create button.



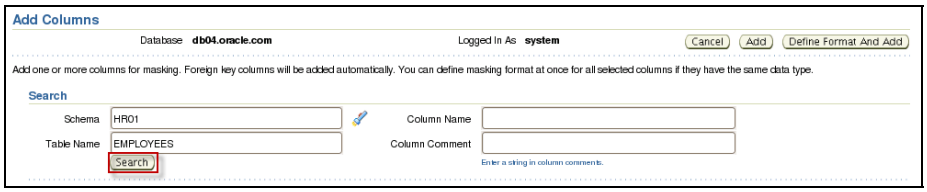
18. From the Create Masking Definition screen, type in the Name, Database and Description field with the provided values below. Continue and click on the Add button.

    Name: DETERMINISTIC_MASKING_EXAMPLE_HR01
    Database: db04.oracle.com
    Description: Sample Deterministic Example

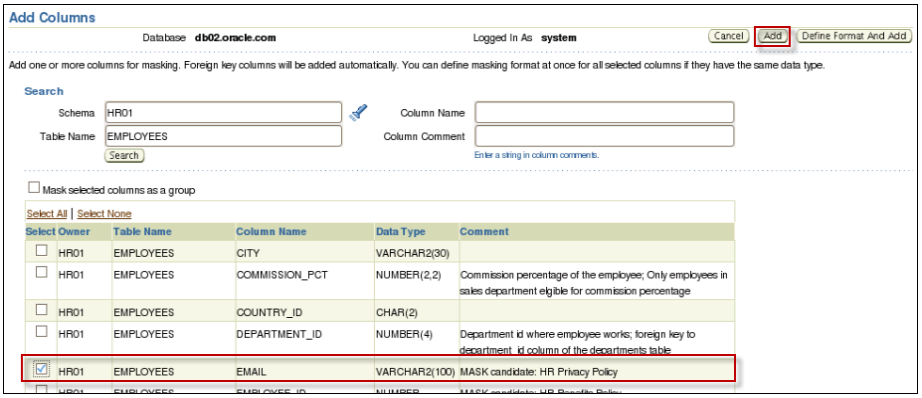


19. In the Add Columns screen, search for the EMPLOYEES table in the HR01 schema. Type in the following values and click on the Search button.

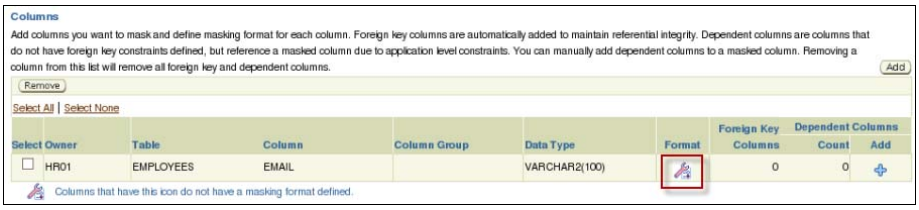
    Schema: HR01
    Table Name: EMPLOYEE



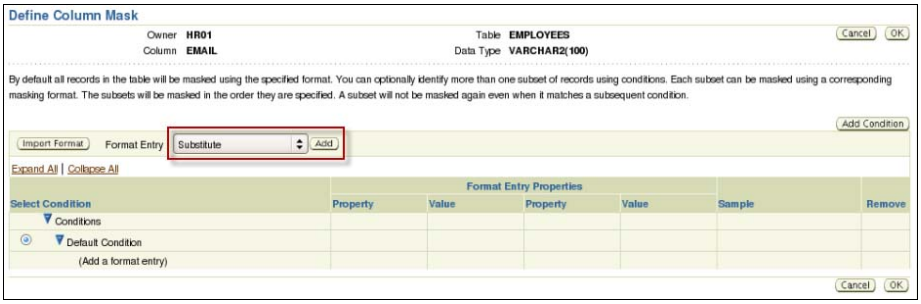
20. Select the column for EMAIL and click on the Add button.



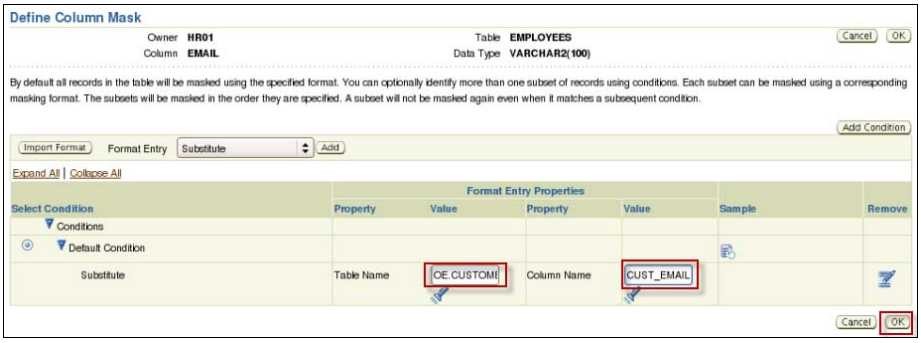
21. Click on the Format icon.



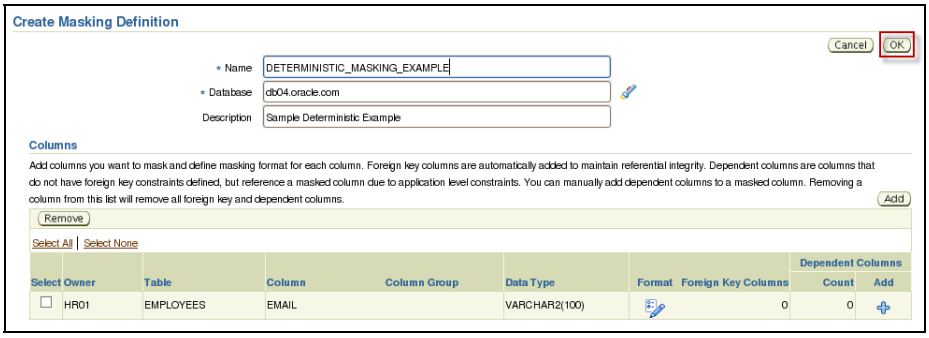
22. In the Define Column Mask section, choose the Format Entry of Substitute and click on the Add button.



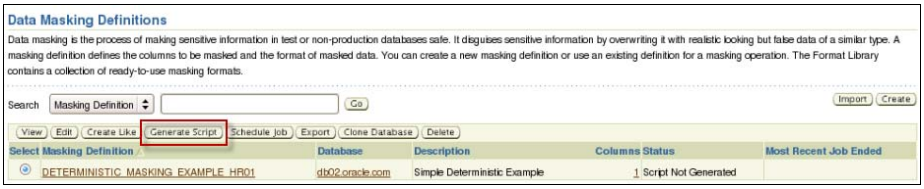
23. Enter the Table Name OE.CUSTOMERS and the Column Name CUST_EMAIL to be used for the substitute values. Click on the OK button to proceed.



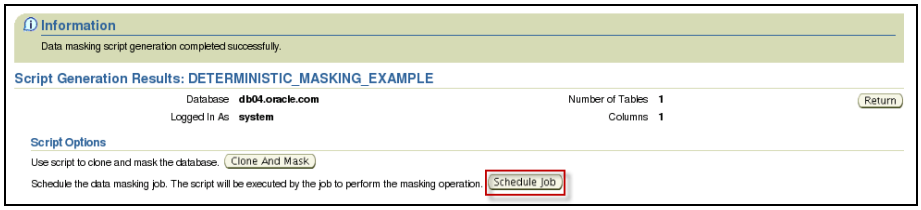
24. After the Masking Definition has been created, click on the OK button.



25. As you have completed in previous exercises, click on the Generate Script button.

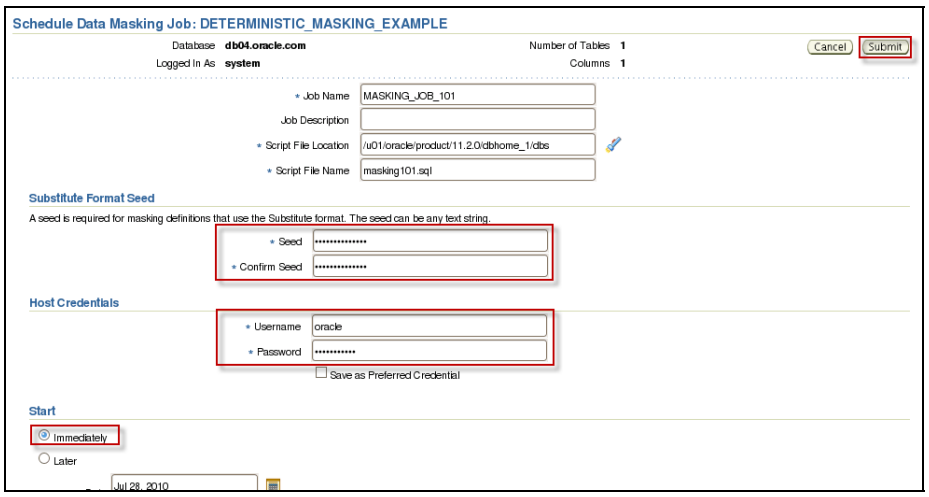


26. After the script has been generated, click on the Schedule Job button.

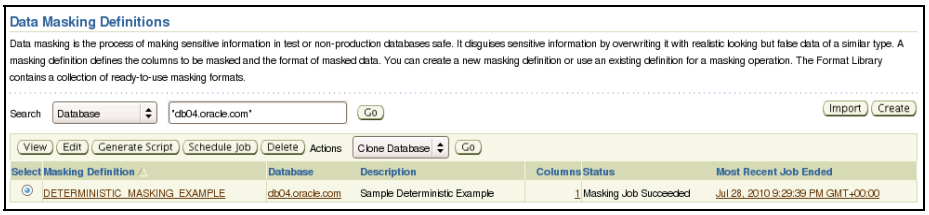


27. Provide the user credentials provided for the oracle user using the password provided (i.e. g0Oracle12#), a Substitute Format Seed (for example, a string “123456”), and click on the Submit button.

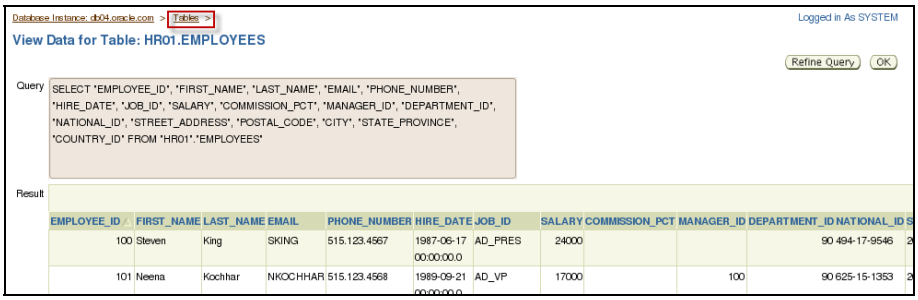
To properly show Deterministic masking in this exercise, you must use the same Seed value for both masking jobs. New in EM 11g, Substitute Format Seeds have been introduced. This allows the user to provide seed values and have better control over non-deterministic and deterministic masking.



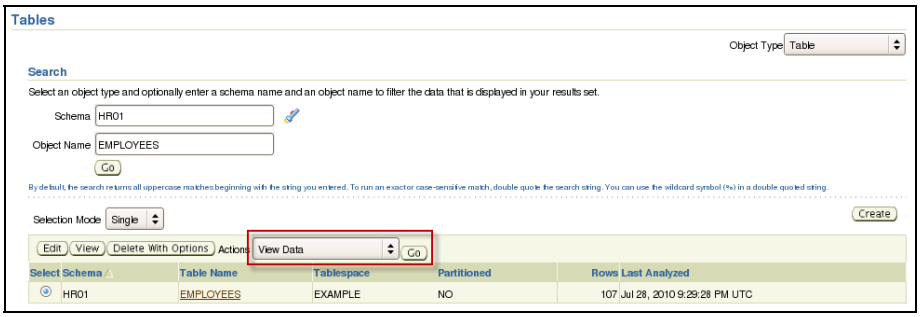
28. One the Masking job is complete, move to the next step.



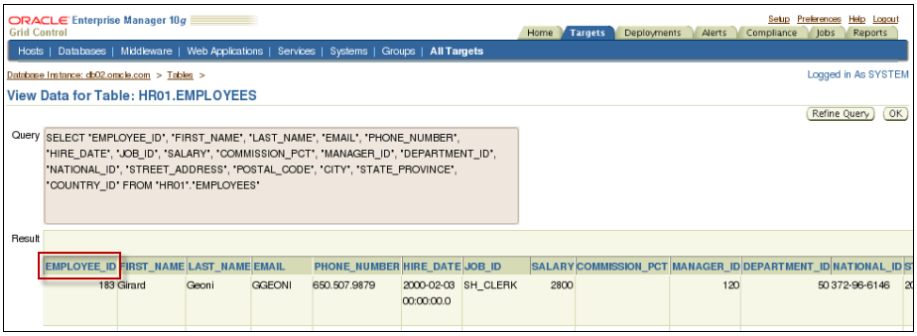
29. In the second browser tab, click on the Tables link.



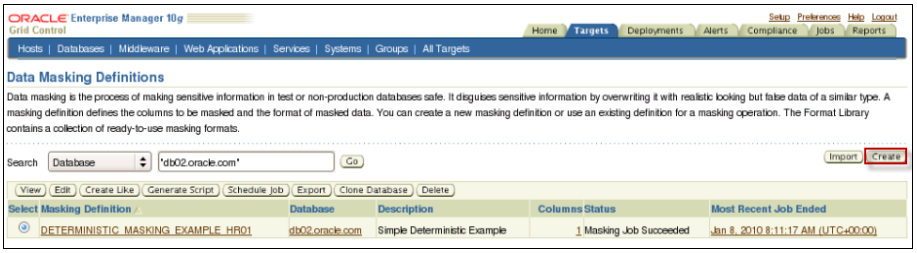
30. Query the HR01 schema and the EMPLOYEES table. Click on the Go button. Select the Action to View Data and click on the Go button. View the masked data.



31. Click on the EMPLOYEE_ID column to sort the data. Keep this tab open. We will now mask the data on HR02 to demonstrate the results of the deterministic masking capability.

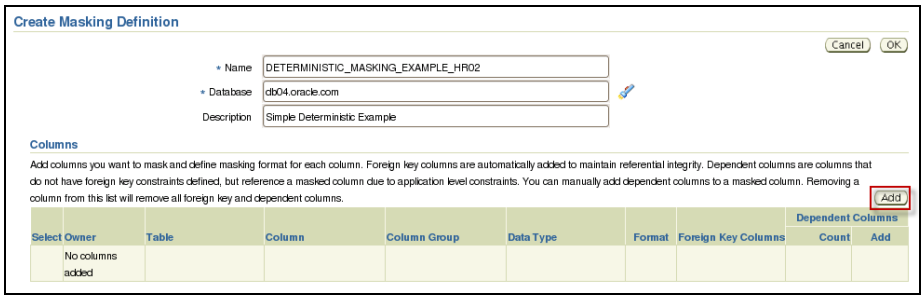


32. Navigate back to the first browser tab. Click on the Create button to create the same masking definition with the only exception being the use of the HR02 table.



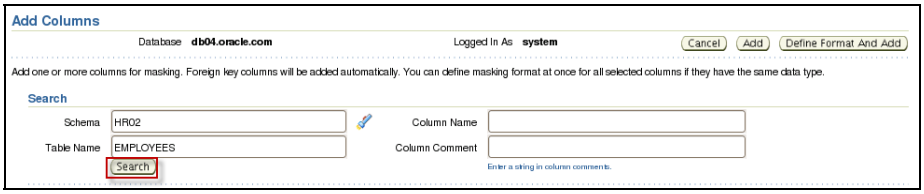
33. From the Create Masking Definition screen, type in the Name, Database and Description field with the provided values below. Continue and click on the Add button.

Name: DETERMINISTIC_MASKING_EXAMPLE_HR02
Database: db04.oracle.com
Description: Sample Deterministic Example

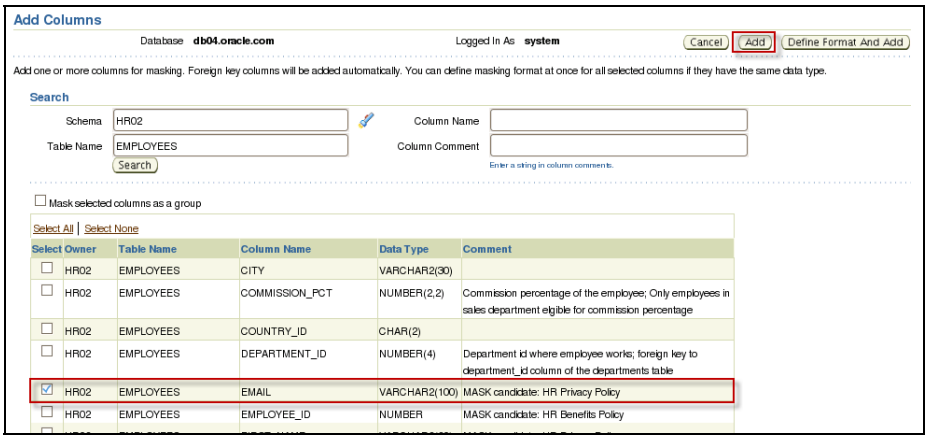


34. In the Add Columns screen, search for the EMPLOYEES table in the HR02 schema. Type in the following values and click on the Search button.

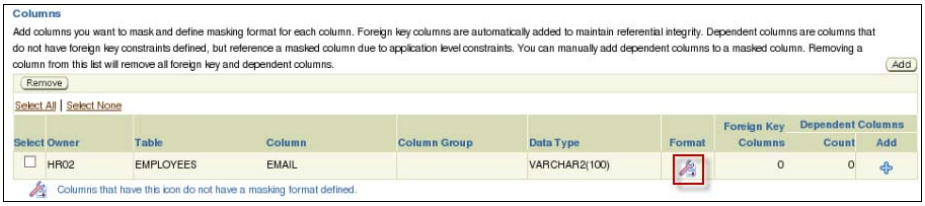
Schema: HR02
Table Name: EMPLOYEE



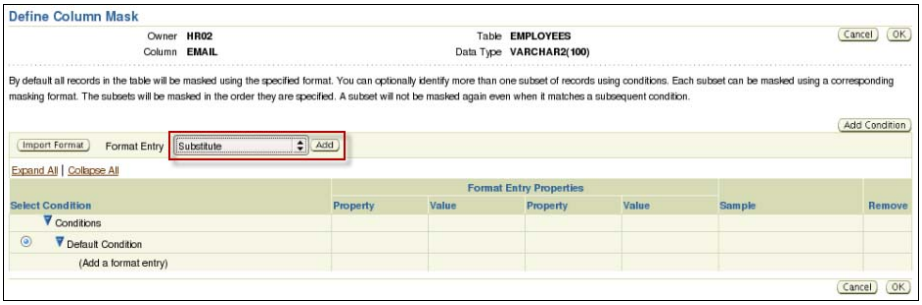
35. Select the column for EMAIL and click on the Add button.



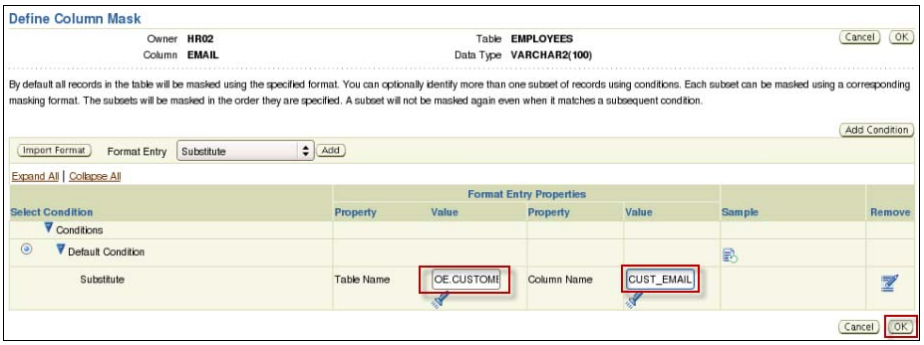
36. Click on the Format icon.



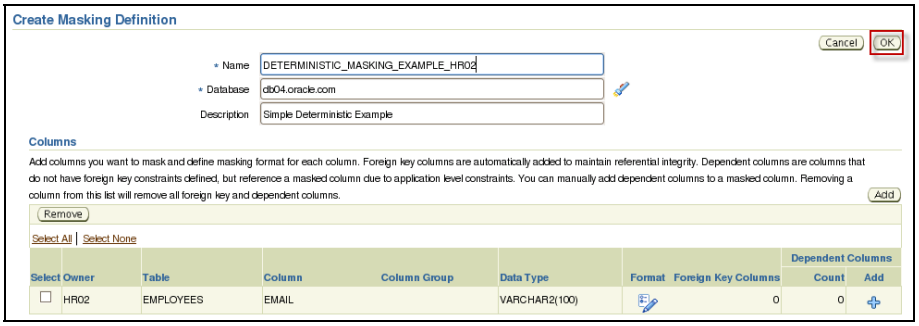
37. In the Define Column Mask section, choose the Format Entry of Substitute and click on the Add button.



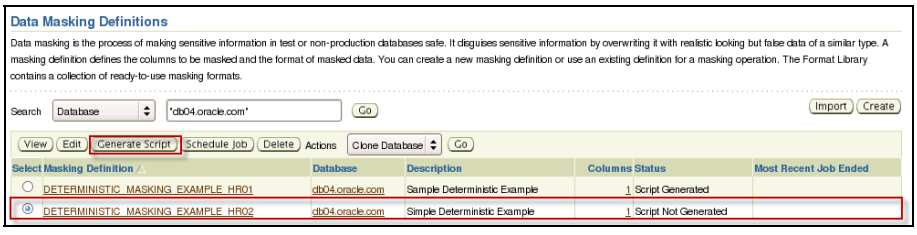
38. Enter the Table Name OE.CUSTOMERS and the Column Name CUST_EMAIL to be used for the substitute values. Click on the OK button to proceed.



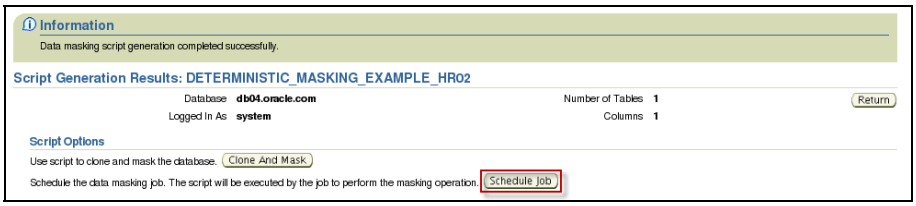
39. After the Masking Definition has been created, click on the OK button.



40. As you have completed in previous exercises, click on the Generate Script button.

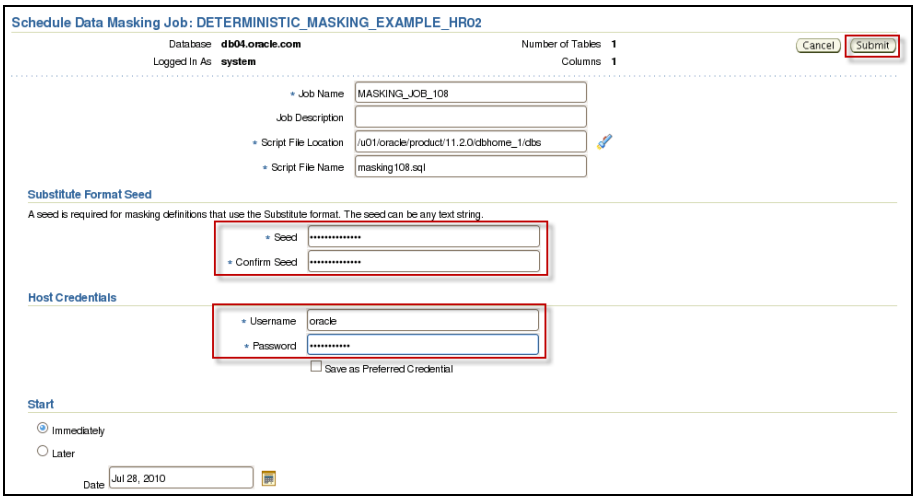


41. After the script has been generated, click on the Schedule Job button.

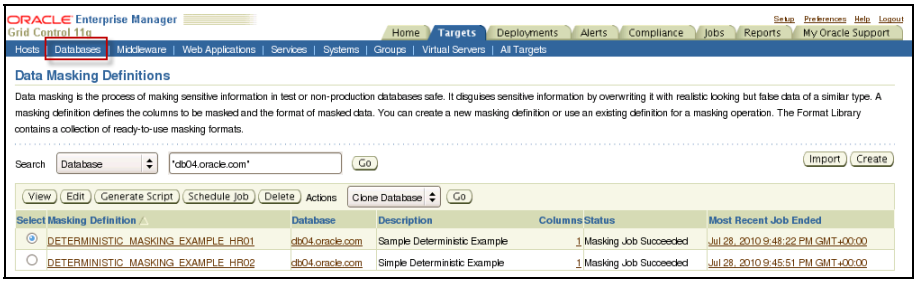


42. Provide the user credentials provided for the oracle user using the password provided (i.e. g0Oracle12#), the same Substitute Format Seed (i.e. seedtextstring) used in the previous step, and click on the Submit button.

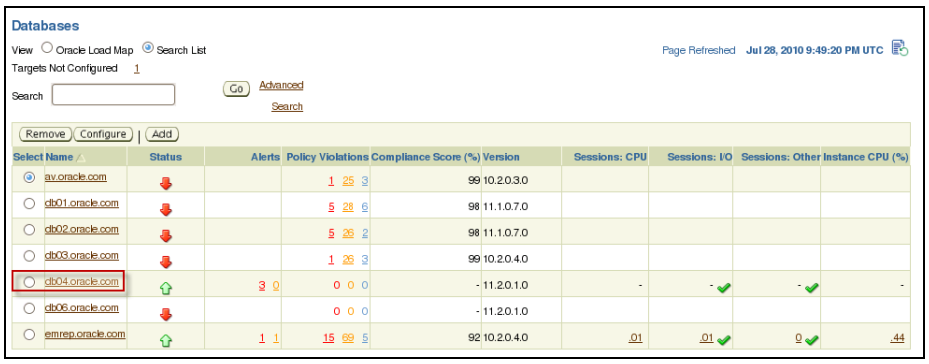
    Again, to properly show Deterministic masking in this exercise, you must use the same Seed value for both masking jobs.



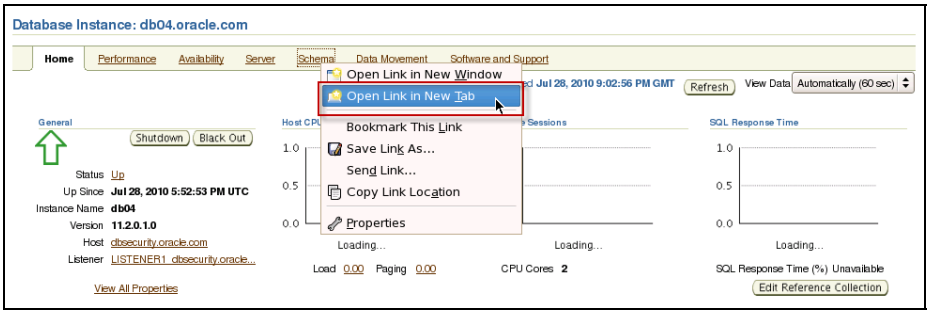
43. Once the masking job is complete, click on Databases link.



44. Click on the database link db04.oracle.com.

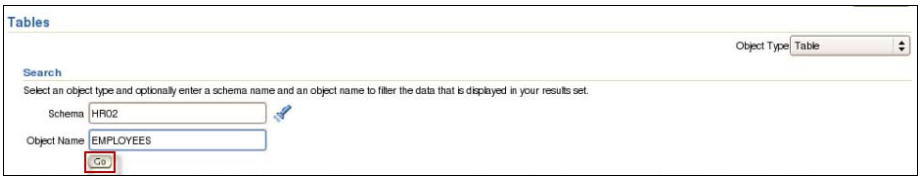


45. Right click on the link, ‘Schema’ and choose the option, ‘Open Link in New Tab’.

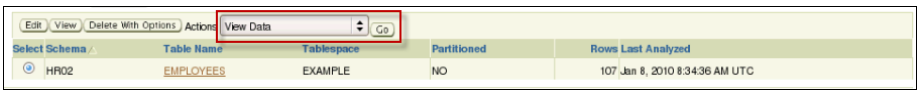


46. In the newly opened tab, we will query the HR02 table. Click on the Tables link.

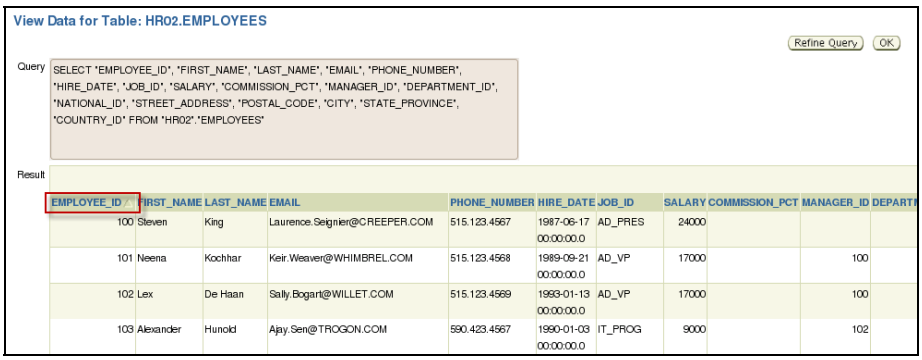
47. We will be querying the HR02 schema and the EMPLOYEES table. Click on the Go button.



48. Select the Action to View Data and click on the Go button.



49. Click on the EMPLOYEE_ID column to sort the data.



50. Compare the two tabs and the results of the masked tables, HR01 and HR02. You will notice the results of deterministic masking. The masked values for the EMAIL column are consistent between these two tables.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL
100	Steven	King	Alexander.Berenger@BECARD.COM
101	Neena	Kochhar	Harrison.Sutherland@GODWIT.COM
102	Lex	De Haan	Rick.Lyon@MERGANSER.COM
103	Alexander	Hunold	Hal.Stockwell@PHOEBE.COM
104	Bruce	Ernst	Shelley.Peckinpah@GODWIT.COM
105	David	Austin	Roy.Hulce@SISKIN.COM
106	Valli	Pataballa	Rosanne.Douglas@ANHINGA.COM
107	Diana	Lorentz	Diane.Mason@TROGON.COM