OFS Performance Analytics Cloud Service

Access Control Guide

Release 9.0.0.0.0

Feb 2020

ORACLE
Financial Services

ORACLE

OFS Performance Analytics Cloud Service

Document Control

Version Number	Revision Date	Change Log
9.0	24 January 2020	First release

Table of Contents

1 Visibility

Visibility is implemented in order to restrict the user's access to the data and the metadata. The user can view based on the role and the privileges assigned to the user.

Visibility has been implemented using two security models:

- OBIEE Security
- Data Security

1.1 OBIEE Security

This has been implemented using the Roles and Privileges settings, the dashboard level, Report level, and the object level.

1.2 Data Security

This has been implemented with a sequence of tables used for controlling the data access to the user.

The set of tables are:

- FSI_M_USER: This table stores all the users who are not relationship managers and are business users who have access to data at different levels. The user id in this table should match the user's login id of OBIEE.
- FSI_M_USER_MANAER_MAP: This table stores all the users who are relation ship managers. V_User_name should hold the Obiee login Id of the user who is a relationship manager. The Manager Code column should match with the entry in dim_management.
- FCT_ACCT_MANAGER_REL: This table restricts the user who is a relationship manager to certain account of customer/Customers. This defines the user at the lowest granularity.
- DIM_CUSTOMER: This table is to define if the user has access to all the accounts the customer holds. This is again to define the relationship manager visbility. This data will be moved from dim_party . Dim_party will be sourced from stg_party_master.
- FSI_USER_DATA_ACCESS: This is a mapper table enabled on AAI Mapper that provides UI for the user to set the visibility. The visibility of the user can be set at the following levels using the mapper - Product, Branch, Legal Entity, and Line of Business.
- FSI_USR_CTRL_ACCESS: This table contains all the records for each user and the access available to the user for every date. The data is sourced from FSI_M_USER_MANAGER_MAP, FSI_USER_DATA_ACCESS, DIM_MANAGEMENT, FCT_COMMON_ACCOUNT_SUMMARY, FCT_ACCT_MANAGER_REL, and DIM_CUSTOMER. The Parent Child hierarchies (derived entities) need to be refreshed before this table load. The names of the hierarchies are MGRPC and CUSTPC. The User has access to all the child nodes in the manager Hierarchy and all the customer hierarchies the user is managing, and the customer hierarchies managed by the child node managers as well.
- CTRLACC: This is a materialized view on the table FSI_USR_CTRL_ACCESS giving the distinct user access to accounts, customers, products, line of business, and legal entity. This view is

used for applying visibility on the rpd. This is created as a derived entity and there is a job to refresh this derived entity.

> **NOTE** Users insertion in FSI_M_USER and FSI_M_USER_MANAGER_MAP has to be done directly into the table. For example, in presence of Single Signon System, these tables need to be loaded with data from single signon system directly.

1.3 Right to be Forgotten

Right to be Forgotten is the task of removing PII (Personally Identifiable Information) of a Data Subject for the given Party. The financial institution can delete PII for those Data Subjects who have requested this Right to be Forgotten functionality.

The Data Subjects may have made significant financial transactions, and/or financial information may be required for regulatory or compliance reporting. Deleting the complete record that consists of PII may lead to issues in data reconciliation. In OFSAA, the PII data will be replaced with randomized values and therefore, the complete Data Subject record is retained. As a result, financial information is retained; however, the associated Party PII is removed permanently.

This section covers the following sub-sections:

1.3.1 Implementation of Right to be Forgotten by OFSAA

To implement Right to be Forgotten:

1. Use the FSI_PARTY_RIGHT_TO_FORGET table to collect the input list of Party IDs for which PII must be removed from the system. The financial institution must source this Party ID list into the FSI_PARTY_RIGHT_TO_FORGET table, and then invoke the batch (<<INFODOM>>_RightToForget) or schedule it.

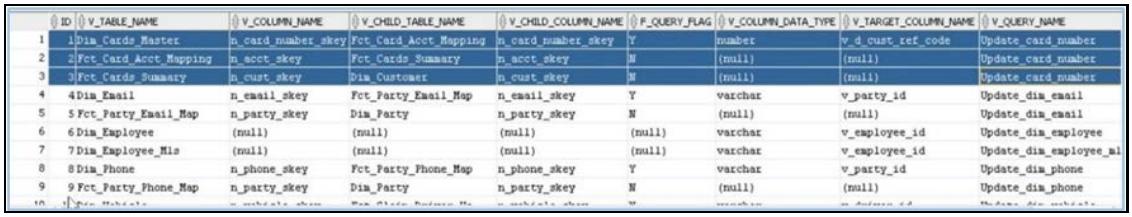
   > **NOTE** For sample query, see Sample Query for the FSI_PARTY_RIGHT_TO_FORGET table.

2. Use the AAI table AAI_DRF_FUNCTION_COLUMN_MAP to store the PII attribute list. During the Right to Forget batch execution, AAI_DRF_FUNCTION_COLUMN_MAP table is referred to randomize the PII values. See the Data Redaction section in OFSAAI Administration Guide.
3. Use the AAI table AAI_DRF_QUERY_METADATA to store the query metadata, which is used during the <<INFODOM>>_RightToForget batch execution. This is the query metadata table that can lead to two types of queries:
   a. When the table consists of Party Identifier as an attribute, a simple record is required in the metadata query table.

For example:

Select v_party_id from Dim_Party where v_party_id='10'

b. When the table does not consist of Party Identifier as an attribute, an interrelated set of records are required in the metadata query table AAI_DRF_QUERY_METADATA. Compose these set of records in a systematic way such that, for the selected Party Identifier, the table join procedure can be performed and traversed to reach the required PII attribute.



	ID	V_TABLE_NAME	V_COLUMN_NAME	V_CHILD_TABLE_NAME	V_CHILD_COLUMN_NAME	F_QUERY_FLAG	V_COLUMN_DATA_TYPE	V_TARGET_COLUMN_NAME	V_QUERY_NAME
1	1	Dim_Cards_Master	n_card_number_skey	Fct_Card_Acct_Mapping	n_card_number_skey	Y	number	v_d_cust_ref_code	Update_card_number
2	2	Fct_Card_Acct_Mapping	n_acct_skey	Fct_Cards_Summary	n_acct_skey	N	(null)	(null)	Update_card_number
3	3	Fct_Cards_Summary	n_cust_skey	Dim_Customer	n_cust_skey	N	(null)	(null)	Update_card_number
4	4	Dim_Email	n_email_skey	Fct_Party_Email_Map	n_email_skey	Y	varchar	v_party_id	Update_dim_email
5	5	Fct_Party_Email_Map	n_party_skey	Dim_Party	n_party_skey	N	(null)	(null)	Update_dim_email
6	6	Dim_Employee	(null)	(null)	(null)	(null)	varchar	v_employee_id	Update_dim_employee
7	7	Dim_Employee_Mis	(null)	(null)	(null)	(null)	varchar	v_employee_id	Update_dim_employee_mi
8	8	Dim_Phone	n_phone_skey	Fct_Party_Phone_Map	n_phone_skey	Y	varchar	v_party_id	Update_dim_phone
9	9	Fct_Party_Phone_Map	n_party_skey	Dim_Party	n_party_skey	N	(null)	(null)	Update_dim_phone

Table definition for AAI_DRF_QUERY_METADATA:

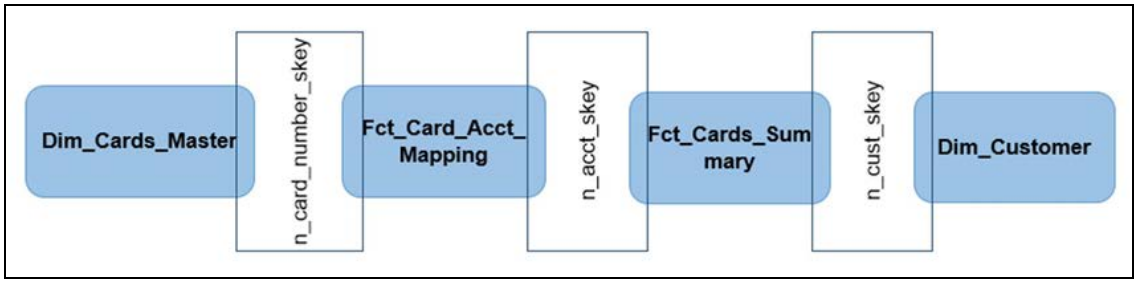
Column Name	Column Type	Description
ID	Number	This is the Primary Key field. You must enter a numerical value.
V_TABLE_NAME	Varchar	This is the source table name.
V_COLUMN_NAME	Varchar	This is the source column name.
V_CHILD_TABLE_NAME	Varchar	This is the table name, which must be linked to the V_TABLE_NAME. If the same table name is repeated with the same column name V_COLUMN_NAME, then the AND condition is formed with V_CHILD_TABLE_NAME. V_CHILD_TABLE_NAME
V_CHILD_COLUMN_NAME	Varchar	This the column name, which must be linked to the V_COLUMN_NAME.
F_QUERY_FLAG	Varchar	Enter Y or N, which is case sensitive. If the value is Y, then you must form a query from V_TABLE_NAME .V_COLUMN_NAME
V_COLUMN_DATA_TYPE	Varchar	Mention the Data Type of the V_COLUMN_NAME. This is required only if F_QUERY_FLAG = Y.
V_TARGET_COLUMN_NAME	Varchar	Mention the PARTY_ID column name, which is required only if F_QUERY_FLAG = Y.
V_QUERY_NAME	Varchar	Mention the same query for a set of joining tables and columns. The set of tables and columns under join query are grouped together using the same query name.

For example:

Dim_Cards_Master table does not consist of n_cust_skey (n_cust_skey is the required Primary Key for the PII Attribute n_card_number_skey). Therefore, perform the table join procedure similar to the following query:

```
Select Dim_Cards_Master.n_card_number_skey from Dim_Cards_Master
Dim_Cards_Master, Fct_Card_Acct_Mapping Fct_Card_Acct_Mapping,

Fct_Cards_Summary Fct_Cards_Summary, Dim_Customer Dim_Customer where
Dim_Cards_Master.n_card_number_skey=Fct_Card_Acct_Mapping.n_card_number
_skey and
Fct_Card_Acct_Mapping.n_acct_skey=Fct_Cards_Summary.n_acct_skey and
Fct_Cards_Summary.n_cust_skey=Dim_Customer.n_cust_skey and
v_d_cust_ref_code='GDPR'
```



Where Dim_Customer.n_cust_skey is a Number Datatype.

> **NOTE** For more sample queries generated using the query metadata table, see Sample Queries using the AAI_DRF_QUERY_METADATA Metadata Table.

To arrive at the above-mentioned query, follow these steps:

In first figure, the required table Dim_Cards_Master does not consist of Party Identifier. Therefore, perform the table join procedure using the AND condition at the table level.

i. Search for a table, which consists of the Party Identifier field. In this query, we have searched for the table Dim_Customer with unique identifier n_cust_skey field. This table must be joined with the required table Dim_Cards_Master.

ii. However, the tables Dim_Cards_Master and Dim_Customer do not consist of any common column name to perform the table join operation. Therefore, search for one more table Fct_Card_Acct_Mapping. This table (Fct_Card_Acct_Mapping) consists of common column name (n_card_number_skey) between Dim_Cards_Master table and itself.

iii. Join the Fct_Card_Acct_Mapping table, which consists of common column name (n_acct_skey) with another table Fct_Cards_Summary.

iv. Join the Fct_Cards_Summary table, which consists of common column name (n_cust_skey) with the final table Dim_Customer.

v. Now, the Dim_Cards_Master table is joined with the Dim_Customer table.

c. You must arrive at the skey or equivalent column in the table, which consists of the required PII attributes. Then the <<INFODOM>>_RightToForget batch uses this key to filter records (For example: Dim_Cards_Master) and randomize all the PIIs listed in the AAI_DRF_FUNCTION_COLUMN_MAP for that table.

4. Now, PII attributes can be queried and the values are randomized.

1.3.2 Sample Queries using the AAI_DRF_QUERY_METADATA Metadata table

These are the sample queries generated using the AAI_DRF_QUERY_METADATA table:

Example 1:

```
select DIM_MANAGEMENT.n_manager_skey from DIM_MANAGEMENT DIM_MANAGEMENT,
FCT_CUSTOMER FCT_CUSTOMER, DIM_CUSTOMER DIM_CUSTOMER where
DIM_MANAGEMENT.n_manager_skey=FCT_CUSTOMER.n_manager_skey and
FCT_CUSTOMER.n_cust_skey=DIM_CUSTOMER.n_cust_skey and
DIM_CUSTOMER.v_d_cust_ref_code in(?,?)
```

Example 2:

```
select  DIM_EMAIL.n_email_skey from DIM_EMAIL DIM_EMAIL, FCT_PARTY_EMAIL_MAP
FCT_PARTY_EMAIL_MAP, DIM_PARTY DIM_PARTY where
DIM_EMAIL.n_email_skey=FCT_PARTY_EMAIL_MAP.n_email_skey and
FCT_PARTY_EMAIL_MAP.n_party_skey=DIM_PARTY.n_party_skey and
DIM_PARTY.v_party_id in(?,?)
```

Example 3:

```
select  STG_CLAIM_DETAILS.v_claim_id from STG_CLAIM_DETAILS
STG_CLAIM_DETAILS, STG_CLAIM_CLAIMANT STG_CLAIM_CLAIMANT where
STG_CLAIM_DETAILS.v_claim_id=STG_CLAIM_CLAIMANT.v_claim_id and
STG_CLAIM_CLAIMANT.v_cust_ref_code in(?,?)
```

Example 4:

```
select STG_CONTACT_MASTER.v_contact_id from STG_CONTACT_MASTER
STG_CONTACT_MASTER, DIM_CONTACT DIM_CONTACT where
STG_CONTACT_MASTER.v_contact_id=DIM_CONTACT.v_contact_id and
DIM_CONTACT.v_customer_id in(?,?)
```

Example 5:

```
select  DIM_CARDS_MASTER.n_card_number_skey from DIM_CARDS_MASTER
DIM_CARDS_MASTER, FCT_CARD_ACCT_MAPPING FCT_CARD_ACCT_MAPPING,
FCT_CARDS_SUMMARY FCT_CARDS_SUMMARY where
DIM_CARDS_MASTER.n_card_number_skey=FCT_CARD_ACCT_MAPPING.n_card_number_skey
and FCT_CARD_ACCT_MAPPING. n_acct_skey=FCT_CARDS_SUMMARY.n_acct_skey and
FCT_CARDS_SUMMARY.v_d_cust_ref_code in(?,?)
```

1.3.3 Sample Query for the FSI_PARTY_RIGHT_TO_FORGET table

This is the sample entry for the FSI_PARTY_RIGHT_TO_FORGET table:

```
Insert into FSI_PARTY_RIGHT_TO_FORGET values (SYSDATE,
<<PARTY_ID_FROM_Ur_ENV>>, 'Testing Right2Forget');
```

1.4 Data Redaction

Data Redaction is one of the Data Security features that provides protection of data against unauthorized access and data theft.

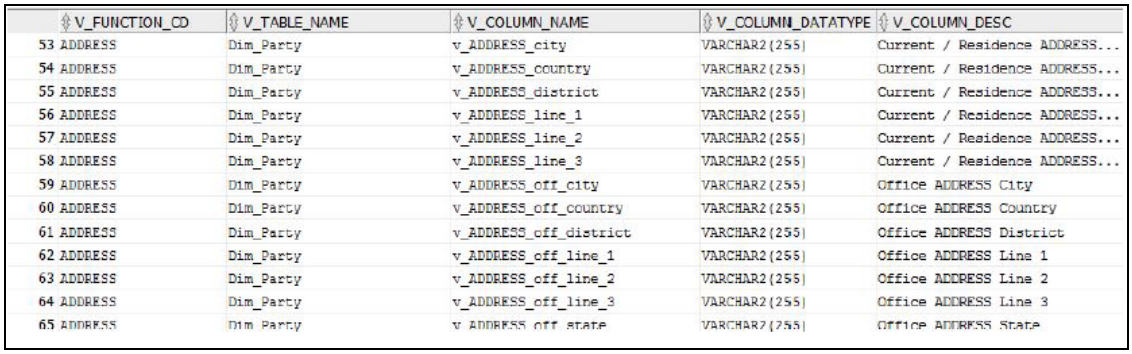
In OFSAA, these tables are seeded as part of Data Redaction:

- AAI_DRF_FUNCTION_MASTER

  This table holds the Redaction function definitions. Generic logical functions can be address, email, card number, phone number etc.

- AAI_DRF_FUNCTION_COLUMN_MAP

  This table holds the Redaction Function- Column mappings. The PII columns will be redacted according to the Function mapping.



	V_FUNCTION_CD	V_TABLE_NAME	V_COLUMN_NAME	V_COLUMN_DATATYPE	V_COLUMN_DESC
53	ADDRESS	Dim_Party	v_ADDRESS_city	VARCHAR2(255)	Current / Residence ADDRESS...
54	ADDRESS	Dim_Party	v_ADDRESS_country	VARCHAR2(255)	Current / Residence ADDRESS...
55	ADDRESS	Dim_Party	v_ADDRESS_district	VARCHAR2(255)	Current / Residence ADDRESS...
56	ADDRESS	Dim_Party	v_ADDRESS_line_1	VARCHAR2(255)	Current / Residence ADDRESS...
57	ADDRESS	Dim_Party	v_ADDRESS_line_2	VARCHAR2(255)	Current / Residence ADDRESS...
58	ADDRESS	Dim_Party	v_ADDRESS_line_3	VARCHAR2(255)	Current / Residence ADDRESS...
59	ADDRESS	Dim_Party	v_ADDRESS_off_city	VARCHAR2(255)	Office ADDRESS City
60	ADDRESS	Dim_Party	v_ADDRESS_off_country	VARCHAR2(255)	Office ADDRESS Country
61	ADDRESS	Dim_Party	v_ADDRESS_off_district	VARCHAR2(255)	Office ADDRESS District
62	ADDRESS	Dim_Party	v_ADDRESS_off_line_1	VARCHAR2(255)	Office ADDRESS Line 1
63	ADDRESS	Dim_Party	v_ADDRESS_off_line_2	VARCHAR2(255)	Office ADDRESS Line 2
64	ADDRESS	Dim_Party	v_ADDRESS_off_line_3	VARCHAR2(255)	Office ADDRESS Line 3
65	ADDRESS	Dim_Party	v_ADDRESS_off_state	VARCHAR2(255)	Office ADDRESS State

- AAI_DRF_TABLE_ACCESS_CD_MAP

  This table holds the mapping of tables having columns marked for redaction to the Access codes. These access codes are SMS function codes and are expected to be mapped to the role DATASECURITY. The policy expression will be created based on this role and evaluated to access non-redacted data.

The list of PIIs, on which Data Redaction is applied, is available at My Oracle Support.

1.4.1 Accessing PII Table and PII Datasheet

- AAI_DRF_FUNCTION_COLUMN_MAP is the PII table.
- PII Datasheet list can be accessed from My Oracle Support.

1.4.2 Data Redaction Batch

Execute the Data Redaction seeded Batch ##INFODOM##_DATA_REDACTION to execute the Data Redaction Utility if it is available as part of application common metadata. If the Batch is not available, you must create a new Batch as mentioned in the Creating Batch for *Executing Data Redaction Utility* section in the OFS Analytical Applications Infrastructure Administration Guide.

The task in the Batch ##INFODOM##_DATA_REDACTION consists of three parameters:

- dataredaction.sh
- true/false
- OFSAA User ID

For more information, see Data Redaction section in the OFS Analytical Applications Infrastructure Administration Guide.

1.4.3 Mapping Roles to User Groups for Data Redaction

Data Controller Group is mapped to DATASECURITYADMIN role:

- Group Code: DATACONTROLLER
- Group Name: Data Controller Group
- Group Description: Data Controller Group
- Role code: DATASECURITYADMIN
- Role Name: Data Security Admin
- Role Description: Data security admin role for executing redaction policies

Mapping from individual applications to DATASECURITY role:

- Role code: DATASECURITY
- Role Name: Data Security Viewer
- Role Description: Data Security Viewer role for viewing original (non-redacted) data.

1. DATASECURITY role must be mapped to those application User Groups which have the privilege to view the data in its originality (un-redacted). Therefore, applications must identify the functions which must be mapped to the DATASECURITY role. These mappings must come as seeded data.
2. And then, map DATASECURITY role to the respective User groups. This mapping must be done manually from individual applications to the DATASECURITY role.

1.4.4 Data Redaction Batch Execution Sample

Data before executing Data Redaction Batch:

Row 1	Fields
N_ACCT_SKEY	6
V_ACCOUNT_NUMBER	BC1007
V_ACCOUNT_DESC	data redaction desc
V_ACCOUNT_MANAGER_CODE	drmc1
V_ORIGINAL_ACCOUNT_NUMBER	data redaction original account numb

Data after executing Data Redaction Batch:

Row 1	Fields
N_ACCT_SKEY	6
V_ACCOUNT_NUMBER	BC1007
V_ACCOUNT_DESC	
V_ACCOUNT_MANAGER_CODE	
V_ORIGINAL_ACCOUNT_NUMBER	

1.5 Data Visibility

Data visibility refers to the data control established on the results fetched by reports depending on the user logged in.

For each user, only those accounts which are directly handled or are handled by a subordinate are visible.

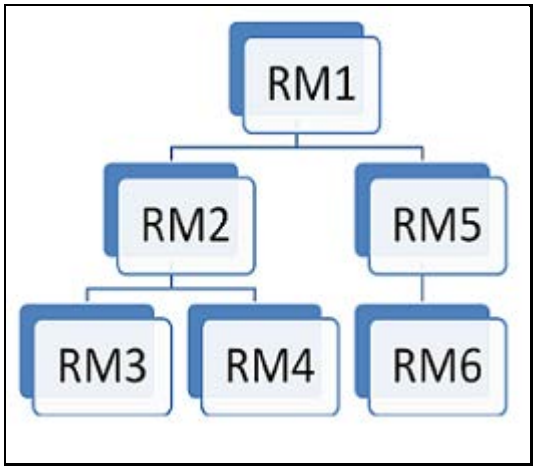
- If the logged in user is a Relationship Management (RM), then only those accounts which are associated to that user's organizational hierarchy will be fetched. This is achieved through FSI_M_USER_MANAGER_MAP table.
- If a user is an RM, then the particular log in ID and the manager code from DIM_MANAGEMENT table have to be populated into FSI_M_USER_MANAGER_MAP table. Also, the table FCT_ACCOUNT_MGR_REL should have the necessary details for the relationship manager to account mapping.

A user logging in without any associated Manager code should have access to the entire data available.

The entries to FSI_M_USER_MANAGER_MAP table have to be manually inserted. It has two columns, V_USERNAME and V_MANAGER_CODE. The V_USERNAME has to be inserted with the login username created in OBIEE and V_MANGER_CODE has to be inserted with the manager code of the corresponding user from DIM_MANAGEMENT table.

Example:

The following diagram depicts a hierarchy of Relationship Managers:



The data visibility for each of the Relationship Managers, starting from the top of the hierarchy is as follows:

- RM1 user has control over the data associated to that user along with the data associated to the immediate subordinates, that is, RM2, RM5, and their subordinates till the end of the hierarchy.
- RM2 user has control over the data associated to that user along with the data associated to the immediate subordinates, that is, RM3, RM4, and their subordinates till the end of the hierarchy.
- RM5 user has control over the data associated to that user along with the data associated to the immediate subordinate, that is, RM6 and his subordinates till the end of the hierarchy.

- If the logged in user is a Sales Representative (SR), the data associated with the opportunities managed by the respective Sales Representatives are visible. Each Sales Representative will have unique skey values and accounts related to those skeys are displayed in reports.

1.6 Report Visibility

This section describes the following sub-sections:

- Report Visibility for IPA
- Report Visibility for RPA

1.6.1 Report Visibility for IPA

The Report's visibility for IPA application is restricted as per the below table:

Application Role	Dashboard	Tab	PII Availability
Business Analyst	Business Summary	Performance Summary Cross Sell Product Performance Line of Business Performance Margin Report Customer Summary What-If Analysis	No No No No No Yes Yes
	Customer Central	Customer 360 Customer Group Customer Performance	Yes Yes Yes
	Opportunities & Activities	Top 10 Opportunities Opportunities Activities	Yes No Yes
Relationship Manager	Relationship Manager Insights	Relationship Manager Insights Top 10 Opportunities Opportunities Activities	Yes Yes No Yes
Administrator	All Dashboards	All Tabs and Reports	Yes

The Reports visibility for the different roles has to be handled by setting proper catalog Permissions. The steps to setup these permissions are described in Setting Up Dashboard Visibility under Configure Roles and Groups of this document.

> **NOTE** Those users who have access to any of the above dashboards with PII columns should also be mapped to the 'Data Security Group' in OFSAA using SMS.
>
> If PII entitlements change for a given user; then you need to either clear the cache through OBIEE admin or refresh the report.

1.6.2 Report Visibility for RPA

The Report's visibility for IPA application is restricted as per the below table:

Application Role	Dashboard	Tab	PII Availability
Business Analyst	Business Analysis	Performance Summary	No
		New Business Analysis	No
		Revenue Analysis	No
		Expense Analysis	No
		Credit Loss Summary	No
		Margin Report	No
		Customer Summary	No
		What-If Analysis	No
	Customer Central	Customer 360	Yes
		Customer Performance	Yes
	Product Summary	All Products	No
		Cards	No
		Retail Bank	No
		Mortgage	No
Relationship Manager	Relationship Manager Insights	Relationship Manager Insights	Yes
Administrator/Super User	All Dashboards	All Tabs and Reports	Yes

The Reports visibility for the different roles has to be handled by setting proper catalog Permissions. The steps to setup these permissions are described in Setting Up Dashboard Visibility under Configure Roles and Groups of this document.

> **NOTE** Those users who have access to any of the above dashboards with PII columns should also be mapped to the 'Data Security Group' in OFSAA using SMS.
>
> If PII entitlements change for a given user; then you need to either clear the cache through OBIEE admin or refresh the report.

1.7 Metadata Visibility

Accessibility to presentation layer objects for creating ad-hoc reports varies from user-to-user depending on the application role the user is allocated.

The following are the requirements for viewing the metadata in RPD:

Application Role	Tables for Ad-hoc Reporting
Business Analyst role	Fact-Account Summary Fact Account Profitability Fact Opportunity Fact Opportunity Activity Fact Management Forecast Fact Customers
Relationship Manager role	Fact Relationship Manager Contribution Fact Relationship Manager Profitability Fact Opportunity Fact Opportunity Activity
Sales Representative role	Fact Opportunity Fact Opportunity Activity Fact Sales Representative Compensation
Administrative role (WebLogic)	All tables