|  |
| --- |
| ***Noetix Security Manager Architecture*** |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Version** | **Comments** |
|  |  |  |  |
| 04/25/19 | Srini Akula | 1.0 | Initial Version |
|  |  |  |  |
|  |  |  |  |

Purpose:

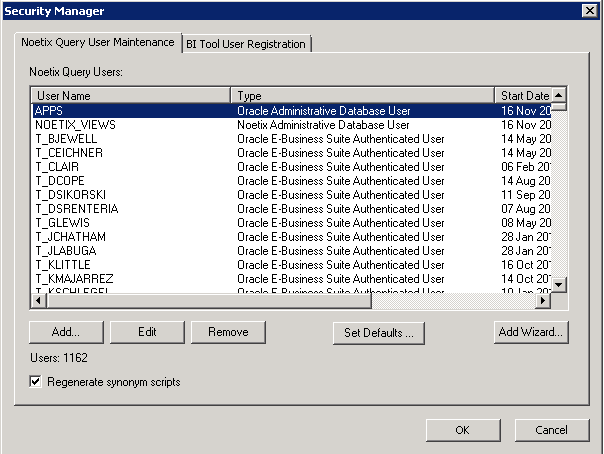
This document contains the architecture of the Noetix Security Manager.

This document provides various security settings done at the Noetix Security Manager window and the tables where this security information is stored.

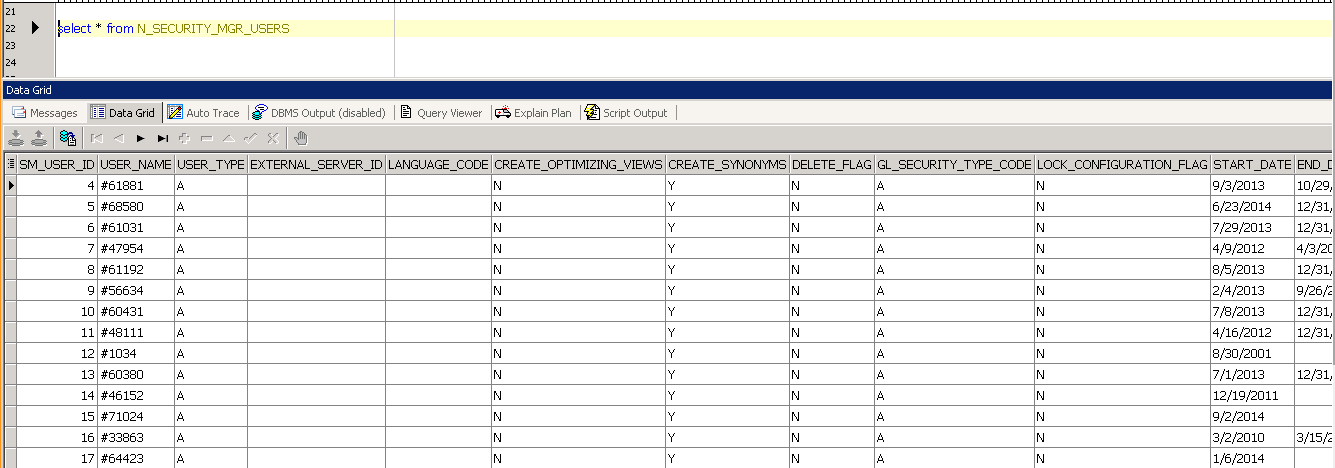
Instance: EBSPJD1

**Query User Registration**

The following screen is used to register Noetix Query Users in Security Manager window.

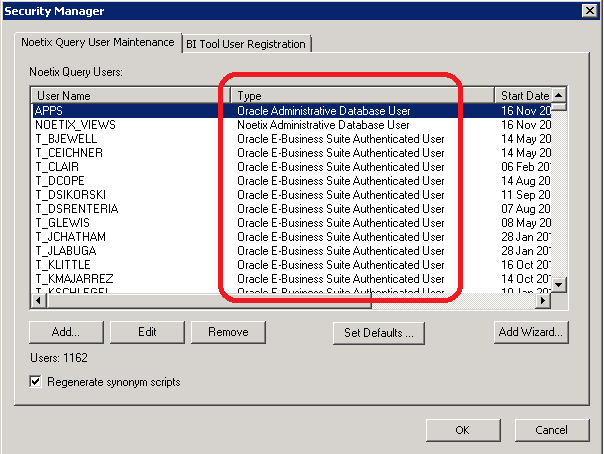


select \* from N\_SECURITY\_MGR\_USERS

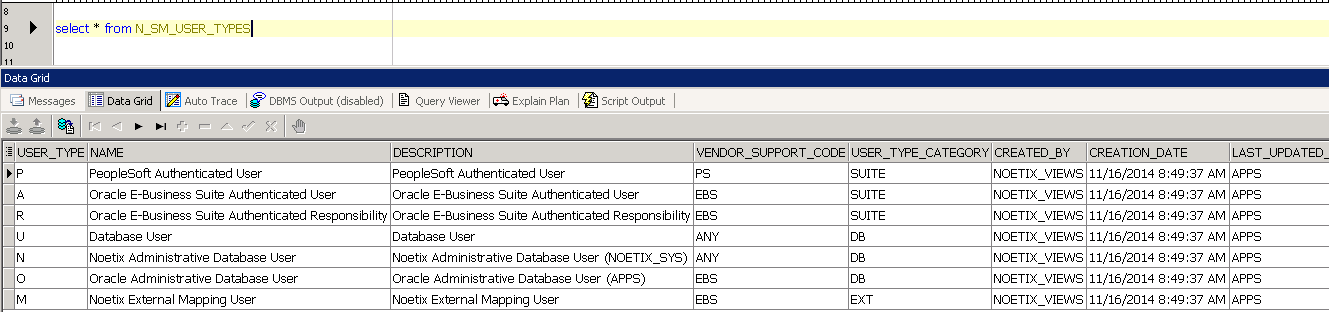


**Query User Types**

The following screen is different types of Noetix Query Users in Security Manager window.

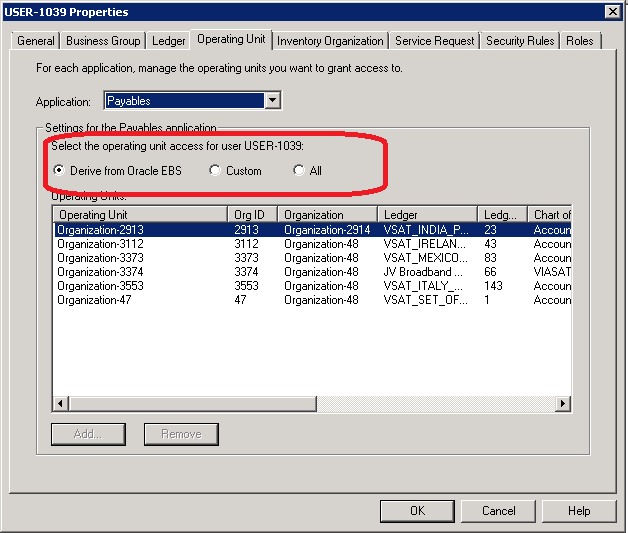


select \* from N\_SM\_USER\_TYPES

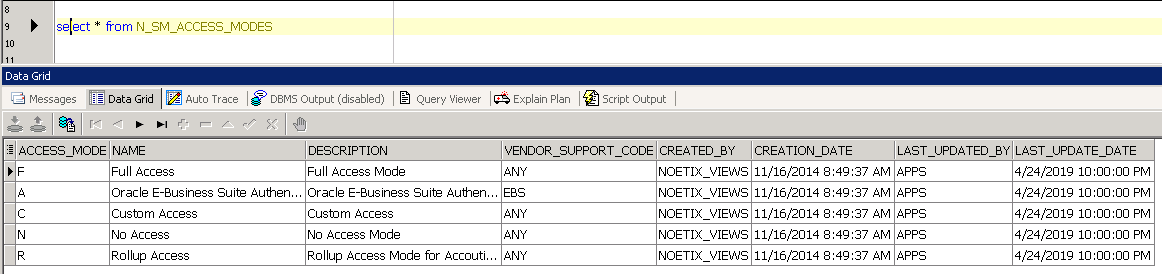


**Access Models**

The following screen is gives us different types of access modes in Security.

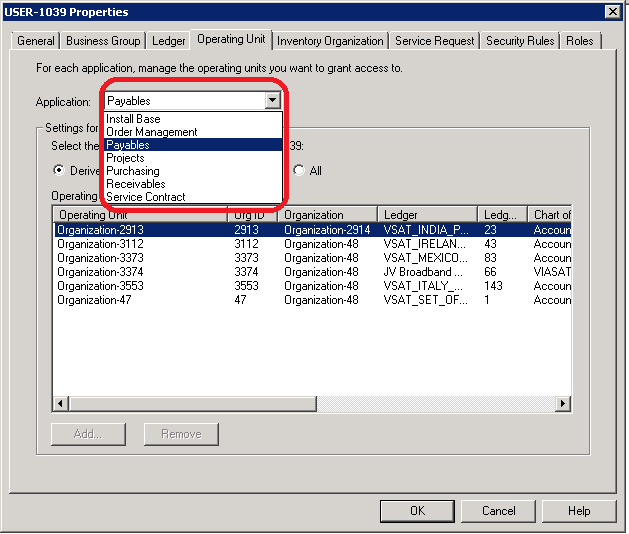


select \* from N\_SM\_ACCESS\_MODES

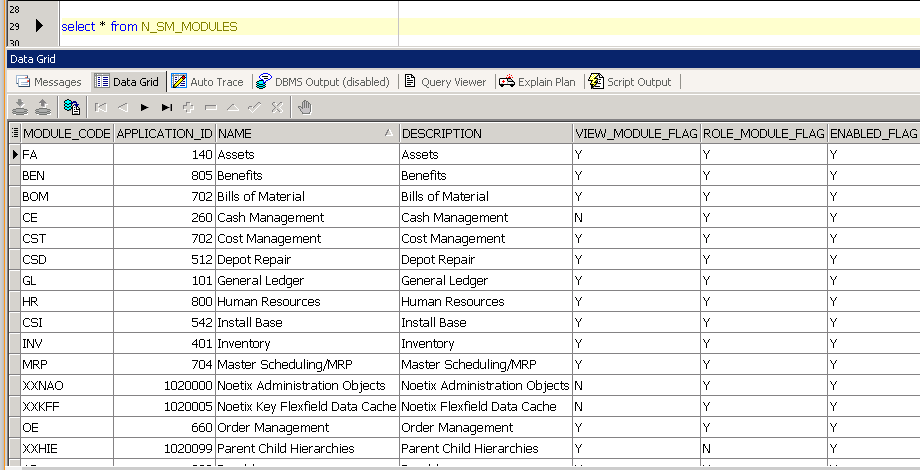


**Modules**

The following screen returns various modules like Payables, Receivables etc., installed in Viasat environment.

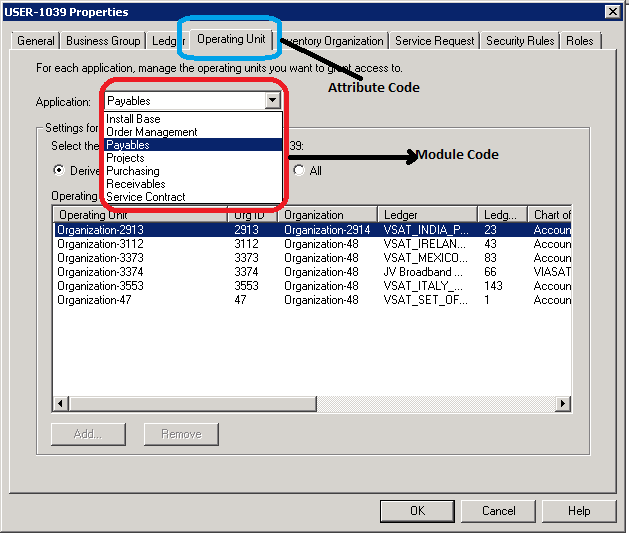


select \* from N\_SM\_MODULES

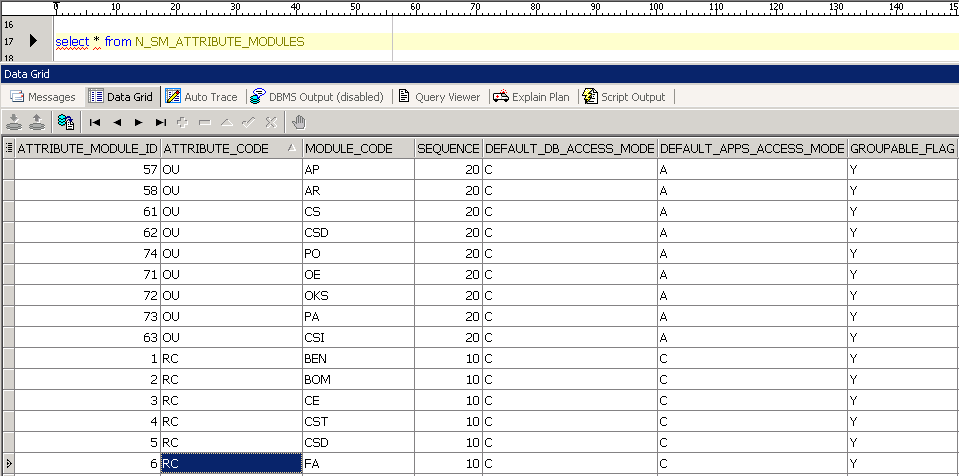


**Attribute Modules**

The following screen returns various types of attributes and its corresponding modules installed in Viasat environment.

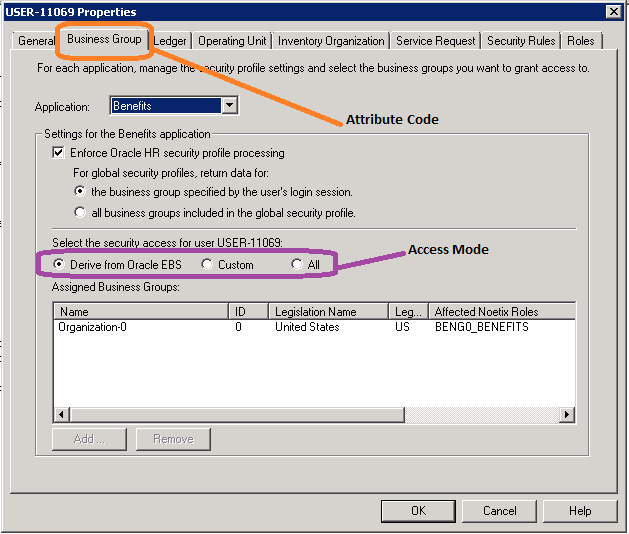


select \* from N\_SM\_ATTRIBUTE\_MODULES

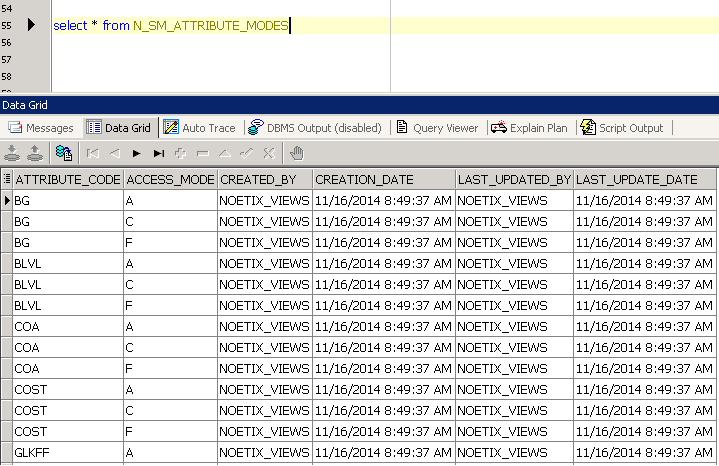


**Attribute Modes**

The following screen returns various attribute codes and its corresponding access modes

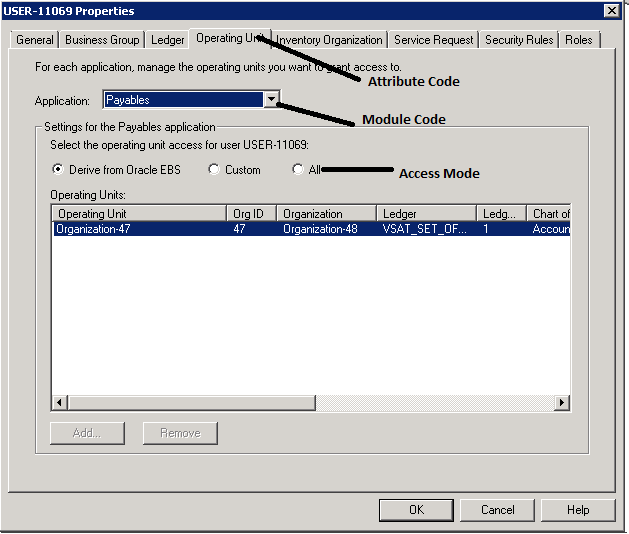


select \* from N\_SM\_ATTRIBUTE\_MODES

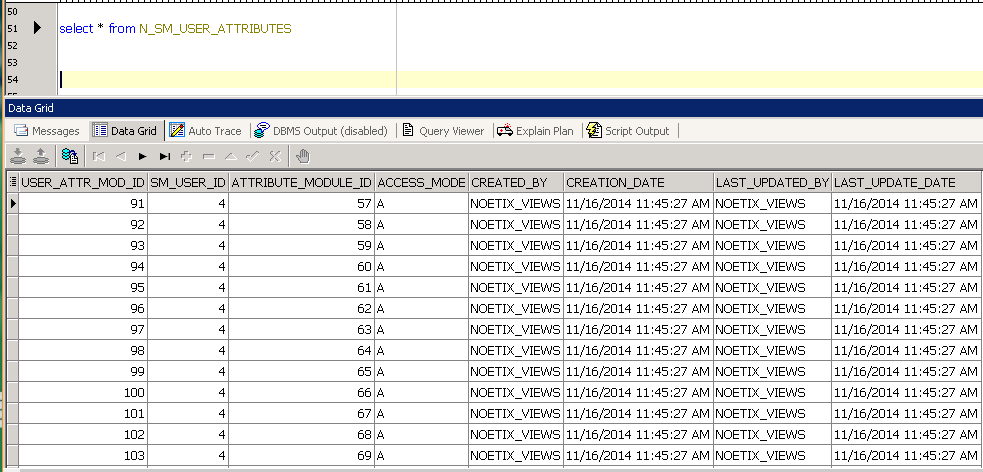


**User Attributes**

The following screen returns various attribute modules and its corressponding access modes.

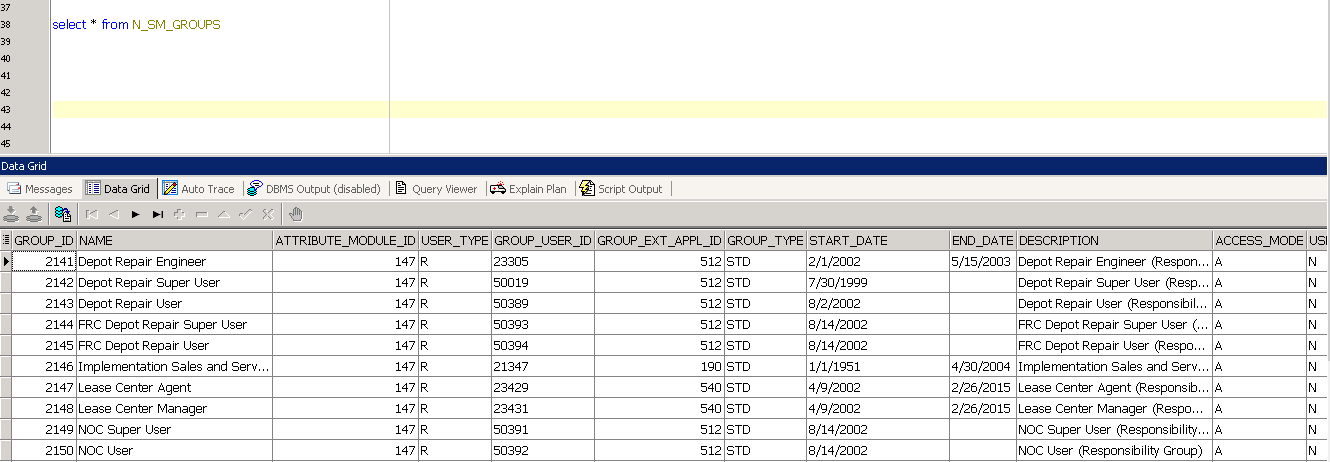


select \* from N\_SM\_USER\_ATTRIBUTES



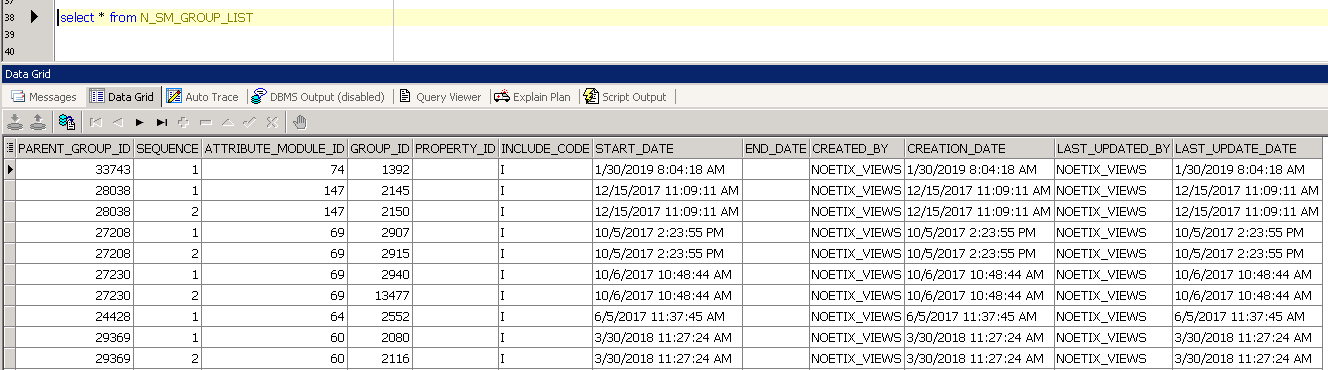
**Groups**

The following screen returns security manger groups. This table consists of all groups like users, responsibilities and access etc.,



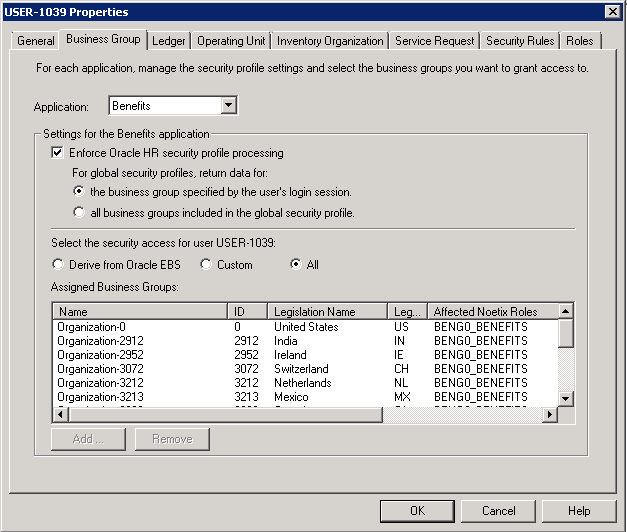
**Group List**

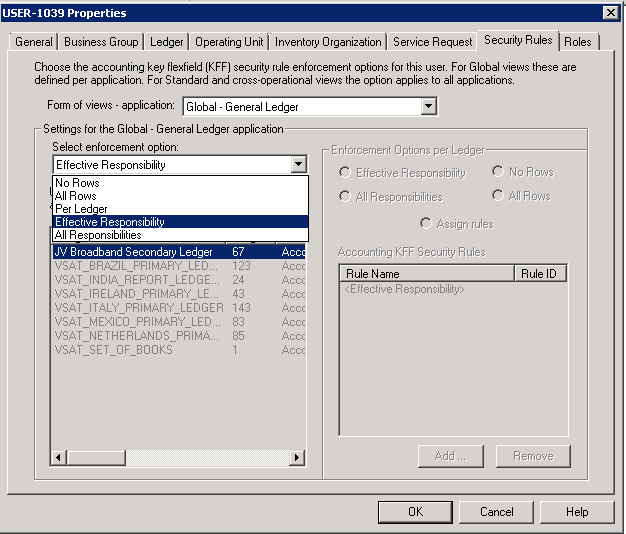
The following screen returns security manger group lists. This table consists relationship and combinational data between groups, properties and attribute modules.



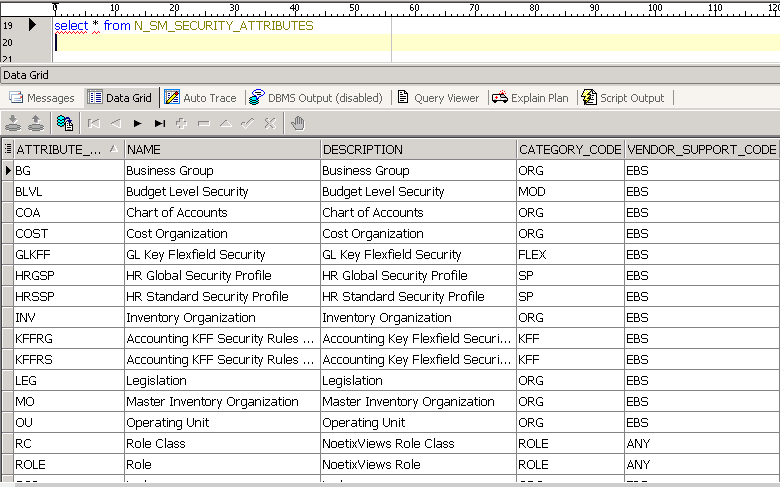
**Security Attributes**

The following screen returns various attributes codes and its security levels.



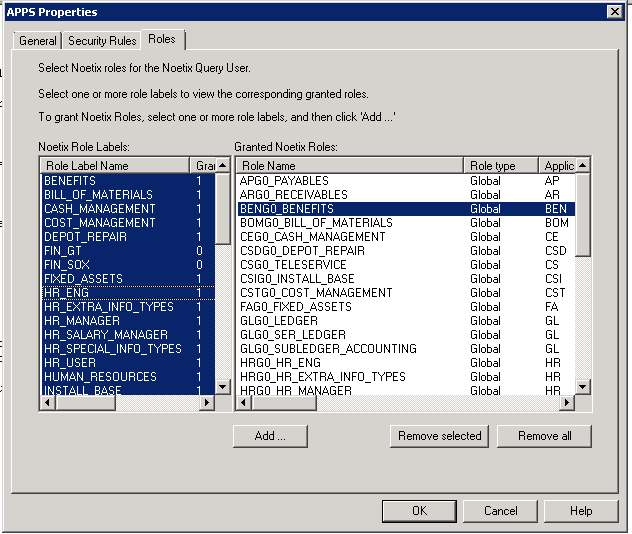


select \* from N\_SM\_SECURITY\_ATTRIBUTES

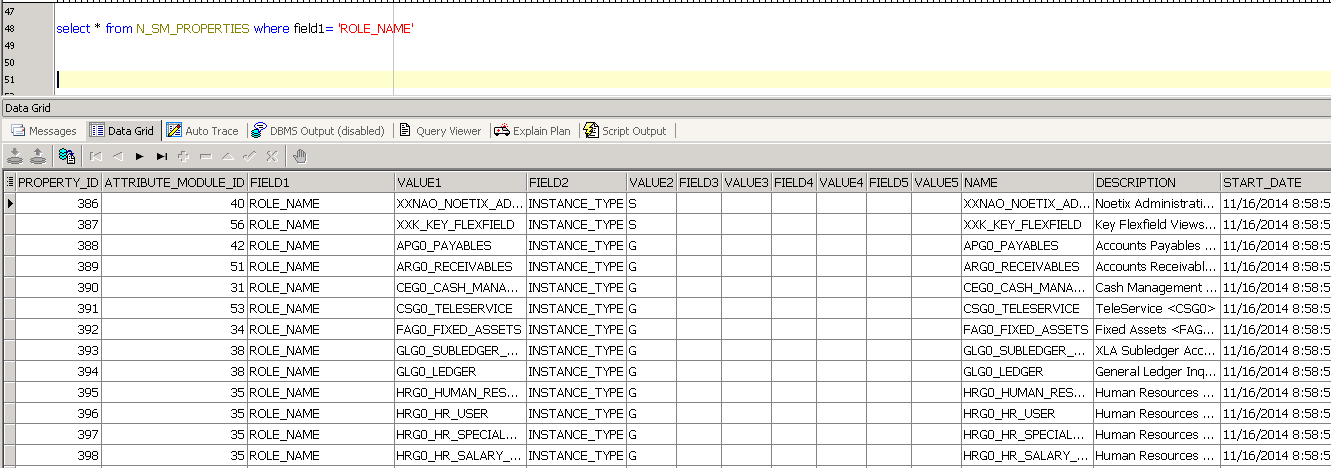


**Properties and Roles**

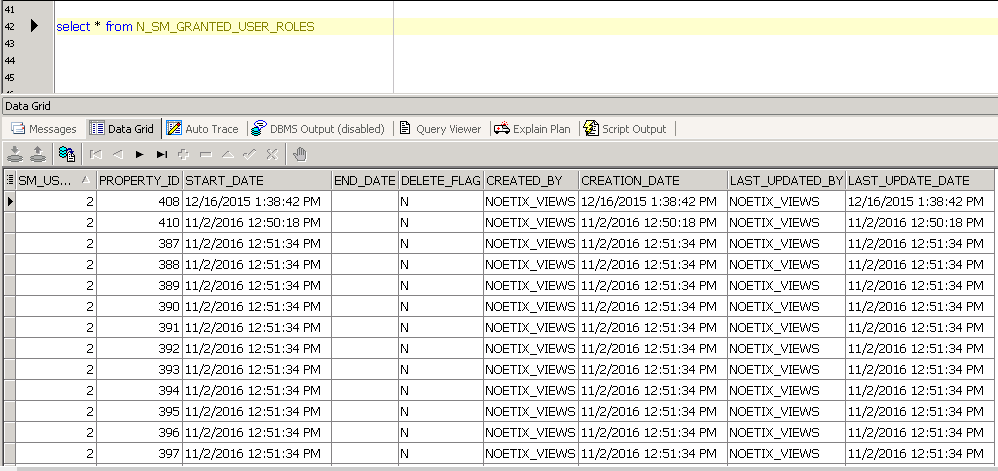
The following screen returns roles assigned to the Noetix Query Users.



select \* from N\_SM\_PROPERTIES where field1= 'ROLE\_NAME'



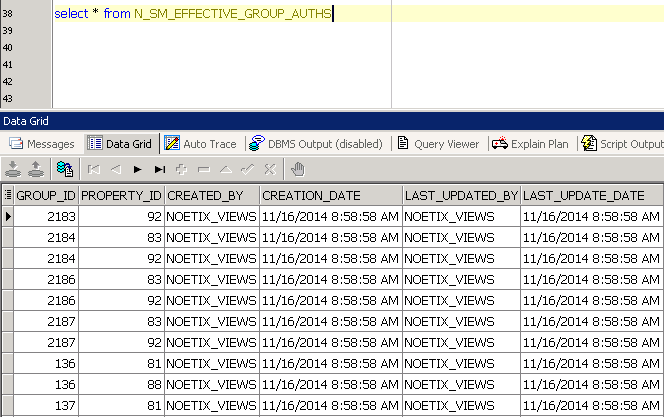
select \* from N\_SM\_GRANTED\_USER\_ROLES



**User Authorizations**

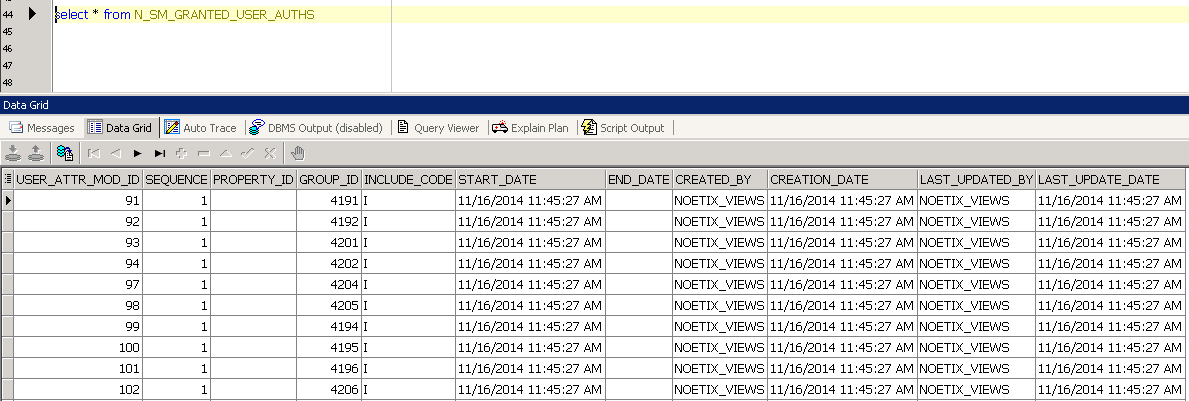
**Group Authorizations**

The group authorizations table stores the relationship between groups and properties.



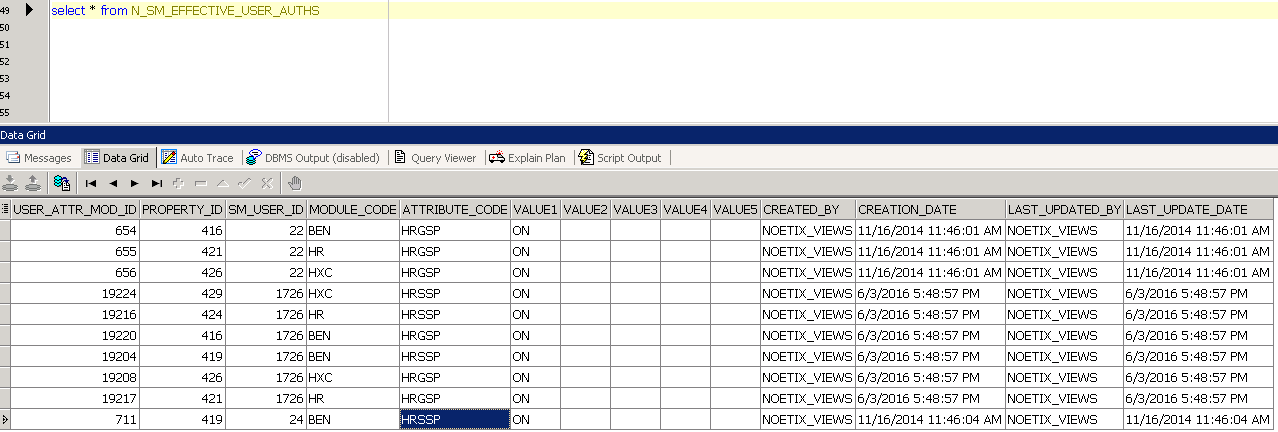
**Granted User Authorizations**

This table stores the relationship between user attribute module, group and properties.



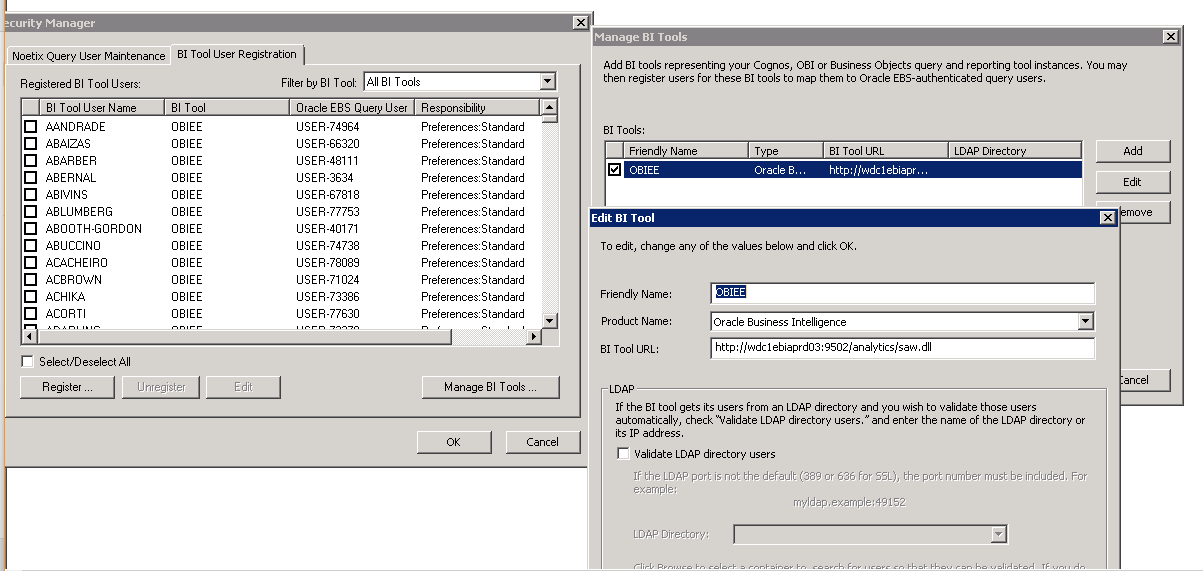
**Effective User Authorizations**

This table stores the relationship between user attribute module, users, properties, modules and access mode values.

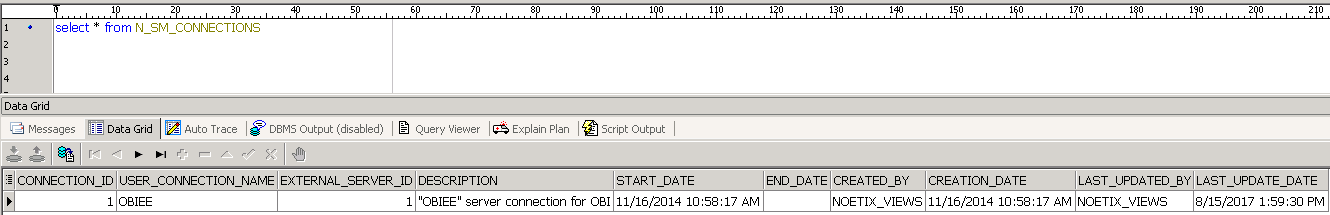


**BI Tool Registrations**

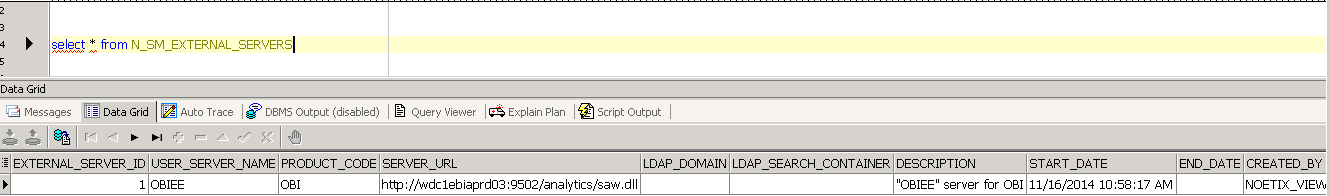
The following screen displays for installing a BI Tool.



select \* from N\_SM\_CONNECTIONS

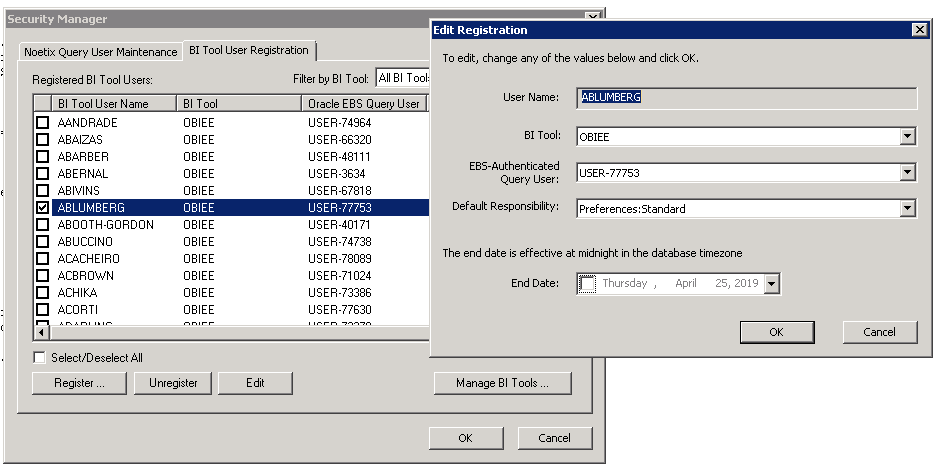


select \* from N\_SM\_EXTERNAL­­\_SERVERS

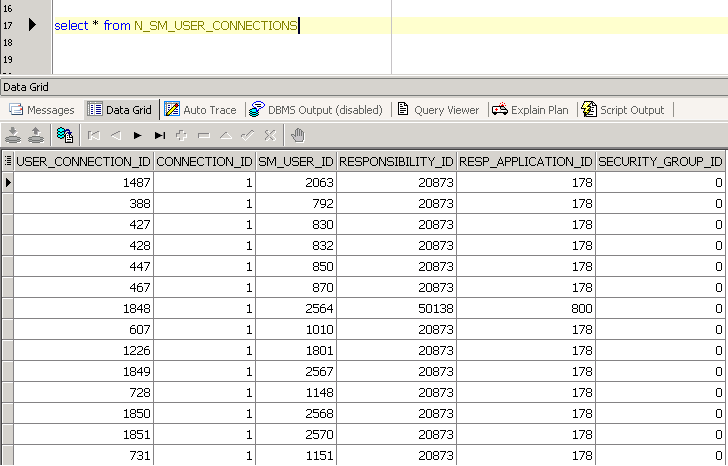


**BI User Registrations**

The following screen displays for registering a BI user.



select \* from N\_SM\_USER\_CONNECTIONS



The below views are created based on the various tables created for Security Manager architecture.

CREATE OR REPLACE FORCE VIEW NOETIX\_VIEWS.N\_SM\_USER\_CONNECTIONS\_V

(

SM\_USER\_ID,

USER\_NAME,

USER\_TYPE,

EXTERNAL\_SERVER\_ID,

EFFECTIVE\_SM\_USER\_ID,

EFFECTIVE\_EXTERNAL\_USER\_NAME,

EFFECTIVE\_USER\_NAME,

EFFECTIVE\_USER\_TYPE,

EFFECTIVE\_FND\_USER\_ID,

CREATE\_OPTIMIZING\_VIEWS,

CREATE\_SYNONYMS,

DELETE\_FLAG,

SELF\_REGISTERED\_FLAG,

USER\_START\_DATE,

USER\_END\_DATE,

USER\_CREATED\_BY,

USER\_CREATION\_DATE,

USER\_LAST\_UPDATED\_BY,

USER\_LAST\_UPDATE\_DATE,

USER\_SERVER\_NAME,

PRODUCT\_CODE,

SERVER\_URL,

LDAP\_DOMAIN,

LDAP\_SEARCH\_CONTAINER,

SERVER\_DESCRIPTION,

SERVER\_START\_DATE,

SERVER\_END\_DATE,

SERVER\_CREATED\_BY,

SERVER\_CREATION\_DATE,

SERVER\_LAST\_UPDATED\_BY,

SERVER\_LAST\_UPDATE\_DATE,

CONNECTION\_ID,

USER\_CONNECTION\_NAME,

CONNECTION\_DESCRIPTION,

CONNECTION\_START\_DATE,

CONNECTION\_END\_DATE,

CONNECTION\_CREATED\_BY,

CONNECTION\_CREATION\_DATE,

CONNECTION\_LAST\_UPDATED\_BY,

CONNECTION\_LAST\_UPDATE\_DATE,

USER\_CONNECTION\_ID,

RESPONSIBILITY\_ID,

RESP\_APPLICATION\_ID,

SECURITY\_GROUP\_ID,

RESPONSIBILITY\_NAME,

SECURITY\_GROUP\_NAME,

RESP\_SEC\_GROUP\_NAME

)

AS

SELECT ext.sm\_user\_id,

ext.user\_name,

ext.user\_type,

ext.external\_server\_id,

ext.effective\_sm\_user\_id,

ext.effective\_external\_user\_name,

ext.effective\_user\_name,

ext.effective\_user\_type,

ext.effective\_fnd\_user\_id,

ext.create\_optimizing\_views,

ext.create\_synonyms,

ext.delete\_flag,

ext.self\_registered\_flag,

ext.user\_start\_date,

ext.user\_end\_date,

ext.user\_created\_by,

ext.user\_creation\_date,

ext.user\_last\_updated\_by,

ext.user\_last\_update\_date,

ext.user\_server\_name,

ext.product\_code,

ext.server\_url,

ext.ldap\_domain,

ext.ldap\_search\_container,

ext.server\_description,

ext.server\_start\_date,

ext.server\_end\_date,

ext.server\_created\_by,

ext.server\_creation\_date,

ext.server\_last\_updated\_by,

ext.server\_last\_update\_date,

ext.connection\_id,

ext.user\_connection\_name,

ext.connection\_description,

ext.connection\_start\_date,

ext.connection\_end\_date,

ext.connection\_created\_by,

ext.connection\_creation\_date,

ext.connection\_last\_updated\_by,

ext.connection\_last\_update\_date,

ext.user\_connection\_id,

ext.responsibility\_id,

ext.resp\_application\_id,

ext.security\_group\_id,

r.responsibility\_name,

s.security\_group\_name,

r.responsibility\_name

|| (CASE APPS.XXNAO\_FND\_WRAPPER\_PKG.Are\_Security\_Groups\_Enabled

WHEN 'Y' THEN ':' || s.security\_group\_name

ELSE TO\_CHAR (NULL)

END)

resp\_sec\_group\_name

FROM n\_fnd\_security\_groups\_vl s,

n\_fnd\_responsibility\_vl r,

n\_sm\_user\_connections\_base\_v ext

WHERE r.responsibility\_id(+) = ext.responsibility\_id

AND r.application\_id(+) = ext.resp\_application\_id

AND s.security\_group\_id(+) = ext.security\_group\_id;

CREATE OR REPLACE FORCE VIEW NOETIX\_VIEWS.N\_SM\_USER\_CONNECTIONS\_BASE\_V

(

SM\_USER\_ID,

USER\_NAME,

USER\_TYPE,

EXTERNAL\_SERVER\_ID,

EFFECTIVE\_SM\_USER\_ID,

EFFECTIVE\_EXTERNAL\_USER\_NAME,

EFFECTIVE\_USER\_NAME,

EFFECTIVE\_USER\_TYPE,

EFFECTIVE\_FND\_USER\_ID,

CREATE\_OPTIMIZING\_VIEWS,

CREATE\_SYNONYMS,

DELETE\_FLAG,

SELF\_REGISTERED\_FLAG,

USER\_START\_DATE,

USER\_END\_DATE,

USER\_CREATED\_BY,

USER\_CREATION\_DATE,

USER\_LAST\_UPDATED\_BY,

USER\_LAST\_UPDATE\_DATE,

USER\_SERVER\_NAME,

PRODUCT\_CODE,

SERVER\_URL,

LDAP\_DOMAIN,

LDAP\_SEARCH\_CONTAINER,

SERVER\_DESCRIPTION,

SERVER\_START\_DATE,

SERVER\_END\_DATE,

SERVER\_CREATED\_BY,

SERVER\_CREATION\_DATE,

SERVER\_LAST\_UPDATED\_BY,

SERVER\_LAST\_UPDATE\_DATE,

CONNECTION\_ID,

USER\_CONNECTION\_NAME,

CONNECTION\_DESCRIPTION,

CONNECTION\_START\_DATE,

CONNECTION\_END\_DATE,

CONNECTION\_CREATED\_BY,

CONNECTION\_CREATION\_DATE,

CONNECTION\_LAST\_UPDATED\_BY,

CONNECTION\_LAST\_UPDATE\_DATE,

USER\_CONNECTION\_ID,

RESPONSIBILITY\_ID,

RESP\_APPLICATION\_ID,

SECURITY\_GROUP\_ID

)

AS

SELECT ext.sm\_user\_id,

ext.user\_name,

ext.user\_type,

ext.external\_server\_id,

ext.effective\_sm\_user\_id,

NVL (eff.external\_user\_name, eff.user\_name)

effective\_external\_user\_name,

eff.user\_name effective\_user\_name,

eff.user\_type effective\_user\_type,

(CASE eff.user\_type WHEN 'A' THEN eff.external\_id ELSE NULL END)

effective\_fnd\_user\_id,

ext.create\_optimizing\_views,

ext.create\_synonyms,

ext.delete\_flag,

ext.self\_registered\_flag,

ext.start\_date user\_start\_date,

ext.end\_date user\_end\_date,

ext.created\_by user\_created\_by,

ext.creation\_date user\_creation\_date,

ext.last\_updated\_by user\_last\_updated\_by,

ext.last\_update\_date user\_last\_update\_date,

c.user\_server\_name,

c.product\_code,

c.server\_url,

c.ldap\_domain,

c.ldap\_search\_container,

c.description server\_description,

c.start\_date server\_start\_date,

c.end\_date server\_end\_date,

c.created\_by server\_created\_by,

c.creation\_date server\_creation\_date,

c.last\_updated\_by server\_last\_updated\_by,

c.last\_update\_date server\_last\_update\_date,

c.connection\_id,

c.user\_connection\_name,

c.connection\_description,

c.connection\_start\_date,

c.connection\_end\_date,

c.connection\_created\_by,

c.connection\_creation\_date,

c.connection\_last\_updated\_by,

c.connection\_last\_update\_date,

u.user\_connection\_id,

u.responsibility\_id,

u.resp\_application\_id,

u.security\_group\_id

FROM n\_sm\_server\_connections\_v c,

n\_sm\_user\_connections u,

n\_security\_mgr\_users eff,

n\_security\_mgr\_users ext

WHERE ext.user\_type = 'M'

AND eff.sm\_user\_id(+) = ext.effective\_sm\_user\_id

AND u.sm\_user\_id = ext.sm\_user\_id

AND c.external\_server\_id = ext.external\_server\_id

AND c.connection\_id = u.connection\_id;

CREATE OR REPLACE FORCE VIEW NOETIX\_VIEWS.N\_SM\_GRANTED\_USER\_ROLES\_V

(

SM\_USER\_ID,

PROPERTY\_ID,

START\_DATE,

END\_DATE,

DELETE\_FLAG,

CREATED\_BY,

CREATION\_DATE,

LAST\_UPDATED\_BY,

LAST\_UPDATE\_DATE,

USER\_NAME,

USER\_TYPE,

USER\_START\_DATE,

USER\_END\_DATE,

USER\_EXTERNAL\_USER\_NAME,

USER\_EXTERNAL\_ID,

USER\_EXTERNAL\_APPL\_ID,

ATTRIBUTE\_CATEGORY\_CODE,

ATTRIBUTE\_CODE,

MODULE\_CODE,

PROPERTY\_ATTRIBUTE\_MODULE\_ID,

ROLE\_NAME,

PROPERTY\_NAME,

PROPERTY\_FIELD1,

PROPERTY\_VALUE1,

PROPERTY\_FIELD2,

PROPERTY\_VALUE2,

PROPERTY\_FIELD3,

PROPERTY\_VALUE3,

PROPERTY\_FIELD4,

PROPERTY\_VALUE4,

PROPERTY\_FIELD5,

PROPERTY\_VALUE5,

PROPERTY\_PARENT\_PROP\_ID,

PROPERTY\_START\_DATE,

PROPERTY\_END\_DATE

)

AS

SELECT ur.sm\_user\_id,

ur.property\_id,

ur.start\_date,

ur.end\_date,

ur.delete\_flag,

ur.created\_by,

ur.creation\_date,

ur.last\_updated\_by,

ur.last\_update\_date,

u.user\_name,

u.user\_type,

u.start\_date user\_start\_date,

u.end\_date user\_end\_date,

u.external\_user\_name user\_external\_user\_name,

u.external\_id user\_external\_id,

u.external\_application\_id user\_external\_appl\_id,

prop.attribute\_category\_code,

prop.attribute\_code,

prop.module\_code,

prop.attribute\_module\_id property\_attribute\_module\_id,

prop.name role\_name,

prop.name property\_name,

prop.field1 property\_field1,

prop.value1 property\_value1,

prop.field2 property\_field2,

prop.value2 property\_value2,

prop.field3 property\_field3,

prop.value3 property\_value3,

prop.field4 property\_field4,

prop.value4 property\_value4,

prop.field5 property\_field5,

prop.value5 property\_value5,

prop.parent\_property\_id property\_parent\_prop\_id,

prop.start\_date property\_start\_date,

prop.end\_date property\_end\_date

FROM n\_sm\_granted\_user\_roles ur,

n\_security\_mgr\_users u,

n\_sm\_properties\_v prop

WHERE ur.sm\_user\_id = u.sm\_user\_id AND ur.property\_id = prop.property\_id;

CREATE OR REPLACE FORCE VIEW NOETIX\_VIEWS.N\_SM\_GRANTED\_USER\_AUTHS\_V

(

USER\_ATTR\_MOD\_ID,

SEQUENCE,

SM\_USER\_ID,

USER\_NAME,

USER\_TYPE,

USER\_START\_DATE,

USER\_END\_DATE,

USER\_FND\_USER\_ID,

USER\_RESPONSIBILITY\_ID,

USER\_RESP\_APPL\_ID,

ATTRIBUTE\_MODULE\_ID,

ATTRIBUTE\_CODE,

ATTRIBUTE\_NAME,

ATTRIBUTE\_DESCRIPTION,

ATTR\_MULTIPLE\_VALUES\_FLAG,

ATTRIBUTE\_ENABLED\_FLAG,

MODULE\_CODE,

MODULE\_NAME,

MODULE\_DESCRIPTION,

MODULE\_ENABLED\_FLAG,

PROPERTY\_ID,

PROPERTY\_ATTRIBUTE\_MODULE\_ID,

PROPERTY\_NAME,

PROPERTY\_FIELD1,

PROPERTY\_VALUE1,

PROPERTY\_FIELD2,

PROPERTY\_VALUE2,

PROPERTY\_FIELD3,

PROPERTY\_VALUE3,

PROPERTY\_FIELD4,

PROPERTY\_VALUE4,

PROPERTY\_FIELD5,

PROPERTY\_VALUE5,

PROPERTY\_PARENT\_PROP\_ID,

GROUP\_ID,

GROUP\_NAME,

GROUP\_TYPE,

GROUP\_ATTRIBUTE\_MODULE\_ID,

GROUP\_START\_DATE,

GROUP\_END\_DATE,

GROUP\_USER\_TYPE,

GROUP\_USER\_ID,

GROUP\_EXT\_APPL\_ID,

GROUP\_USER\_NAME,

FND\_USER\_ID,

RESPONSIBILITY\_APPLICATION\_ID,

RESPONSIBILITY\_ID,

GROUP\_USER\_EDITABLE\_FLAG,

INCLUDE\_CODE,

START\_DATE,

END\_DATE

)

AS

SELECT auth.user\_attr\_mod\_id,

auth.sequence,

uattr.sm\_user\_id,

usr.user\_name,

usr.user\_type,

usr.start\_date user\_start\_date,

usr.end\_date user\_end\_date,

(CASE usr.user\_type WHEN 'A' THEN usr.external\_id ELSE NULL END)

user\_fnd\_user\_id,

(CASE usr.user\_type WHEN 'R' THEN usr.external\_id ELSE NULL END)

user\_responsibility\_id,

usr.external\_application\_id user\_resp\_appl\_id,

uattr.attribute\_module\_id,

am.attribute\_code,

am.attribute\_name,

am.attribute\_description,

am.attr\_multiple\_values\_flag,

am.attribute\_enabled\_flag,

am.module\_code,

am.module\_name,

am.module\_description,

am.module\_enabled\_flag,

auth.property\_id,

prop.attribute\_module\_id property\_attribute\_module\_id,

prop.name property\_name,

prop.field1 property\_field1,

prop.value1 property\_value1,

prop.field2 property\_field2,

prop.value2 property\_value2,

prop.field3 property\_field3,

prop.value3 property\_value3,

prop.field4 property\_field4,

prop.value4 property\_value4,

prop.field5 property\_field5,

prop.value5 property\_value5,

prop.parent\_property\_id property\_parent\_prop\_id,

auth.GROUP\_ID,

grp.name group\_name,

grp.group\_type group\_type,

grp.attribute\_module\_id group\_attribute\_module\_id,

grp.start\_date group\_start\_date,

grp.end\_date group\_end\_date,

grp.user\_type GROUP\_USER\_TYPE,

grp.group\_user\_id,

grp.group\_ext\_appl\_id,

(CASE grp.user\_type WHEN 'U' THEN grp.group\_user\_id ELSE NULL END)

GROUP\_USER\_NAME,

(CASE grp.user\_type WHEN 'A' THEN grp.group\_user\_id ELSE NULL END)

fnd\_user\_id,

(CASE grp.user\_type

WHEN 'R' THEN grp.group\_ext\_appl\_id

ELSE NULL

END)

responsibility\_application\_id,

(CASE grp.user\_type WHEN 'R' THEN grp.group\_user\_id ELSE NULL END)

responsibility\_id,

grp.user\_editable\_flag group\_user\_editable\_flag,

auth.include\_code,

auth.start\_date,

auth.end\_date

FROM n\_sm\_groups grp,

n\_sm\_properties prop,

n\_sm\_attribute\_modules\_v am,

n\_security\_mgr\_users usr,

n\_sm\_user\_attributes uattr,

n\_sm\_granted\_user\_auths auth

WHERE uattr.user\_attr\_mod\_id = auth.user\_attr\_mod\_id

AND usr.sm\_user\_id = uattr.sm\_user\_id

AND usr.user\_type <> 'M'

AND am.attribute\_module\_id = uattr.attribute\_module\_id

AND prop.property\_id(+) = auth.property\_id

AND grp.GROUP\_ID(+) = auth.GROUP\_ID;

CREATE OR REPLACE FORCE VIEW NOETIX\_VIEWS.N\_SM\_EFF\_USER\_GEN\_AUTHS\_V

(

USER\_ATTR\_MOD\_ID,

ACCESS\_MODE,

SEQUENCE,

INCLUDE\_CODE,

SM\_USER\_ID,

ATTRIBUTE\_MODULE\_ID,

ATTRIBUTE\_CODE,

MODULE\_CODE,

CATEGORY\_CODE,

MULTIPLE\_VALUES\_FLAG,

GROUP\_ID,

GROUP\_NAME,

GROUP\_USER\_TYPE,

GROUP\_USER\_ID,

GROUP\_EXT\_APPL\_ID,

GROUP\_TYPE,

PROPERTY\_ID,

PROPERTY\_NAME,

FIELD1,

VALUE1,

FIELD2,

VALUE2,

FIELD3,

VALUE3,

FIELD4,

VALUE4,

FIELD5,

VALUE5,

PARENT\_PROPERTY\_ID

)

AS

SELECT ua.user\_attr\_mod\_id,

ua.access\_mode,

ugrnt.sequence,

ugrnt.include\_code,

ua.sm\_user\_id,

ua.attribute\_module\_id,

am.attribute\_code,

am.module\_code,

attr.category\_code,

attr.multiple\_values\_flag,

grp.GROUP\_ID,

grp.name group\_name,

grp.user\_type group\_user\_type,

grp.group\_user\_id,

grp.GROUP\_EXT\_APPL\_ID,

grp.group\_type,

prop.property\_id,

prop.name property\_name,

prop.field1,

prop.value1,

prop.field2,

prop.value2,

prop.field3,

prop.value3,

prop.field4,

prop.value4,

prop.field5,

prop.value5,

prop.parent\_property\_id

FROM n\_sm\_properties prop,

n\_sm\_effective\_group\_auths gauth,

n\_sm\_groups grp,

n\_sm\_granted\_user\_auths ugrnt,

n\_sm\_modules MOD,

n\_sm\_security\_attributes attr,

n\_sm\_attribute\_modules am,

n\_sm\_user\_attributes ua

WHERE am.attribute\_module\_id = ua.attribute\_module\_id

AND attr.attribute\_code = am.attribute\_code

AND (CASE am.attribute\_code

WHEN 'KFFRG'

THEN

(CASE grp.access\_mode WHEN 'N' THEN '-999.999' WHEN 'F' THEN '-888.888' END)

WHEN 'KFFRS'

THEN

(CASE grp.access\_mode WHEN 'N' THEN '-999.999' WHEN 'F' THEN '-888.888' END)

ELSE

prop.value1

END) = prop.value1

AND NVL (attr.enabled\_flag, 'N') = 'Y'

AND MOD.module\_code = am.module\_code

AND NVL (MOD.enabled\_flag, 'N') = 'Y'

AND ugrnt.user\_attr\_mod\_id = ua.user\_attr\_mod\_id

AND SYSDATE BETWEEN ugrnt.start\_date

AND NVL (ugrnt.end\_date, SYSDATE + 1)

AND grp.GROUP\_ID = ugrnt.GROUP\_ID

AND SYSDATE BETWEEN grp.start\_date

AND NVL (grp.end\_date, SYSDATE + 1)

AND gauth.GROUP\_ID = grp.GROUP\_ID

AND prop.property\_id = gauth.property\_id

AND SYSDATE BETWEEN prop.start\_date

AND NVL (prop.end\_date, SYSDATE + 1)

UNION ALL

SELECT ua.user\_attr\_mod\_id,

ua.access\_mode,

ugrnt.sequence,

ugrnt.include\_code,

ua.sm\_user\_id,

ua.attribute\_module\_id,

am.attribute\_code,

am.module\_code,

attr.category\_code,

attr.multiple\_values\_flag,

NULL GROUP\_ID,

NULL group\_name,

NULL group\_user\_type,

NULL group\_user\_id,

NULL GROUP\_EXT\_APPL\_ID,

NULL group\_type,

prop.property\_id,

prop.name property\_name,

prop.field1,

prop.value1,

prop.field2,

prop.value2,

prop.field3,

prop.value3,

prop.field4,

prop.value4,

prop.field5,

prop.value5,

prop.parent\_property\_id

FROM n\_sm\_properties prop,

n\_sm\_granted\_user\_auths ugrnt,

n\_sm\_modules MOD,

n\_sm\_security\_attributes attr,

n\_sm\_attribute\_modules am,

n\_sm\_user\_attributes ua

WHERE am.attribute\_module\_id = ua.attribute\_module\_id

AND attr.attribute\_code = am.attribute\_code

AND NVL (attr.enabled\_flag, 'N') = 'Y'

AND MOD.module\_code = am.module\_code

AND NVL (MOD.enabled\_flag, 'N') = 'Y'

AND ugrnt.user\_attr\_mod\_id = ua.user\_attr\_mod\_id

AND SYSDATE BETWEEN ugrnt.start\_date

AND NVL (ugrnt.end\_date, SYSDATE + 1)

AND prop.property\_id = ugrnt.property\_id

AND SYSDATE BETWEEN prop.start\_date

AND NVL (prop.end\_date, SYSDATE + 1);

CREATE OR REPLACE FORCE VIEW NOETIX\_VIEWS.N\_SM\_EFF\_USER\_AUTHS\_V

(

USER\_ATTR\_MOD\_ID,

PROPERTY\_ID,

SM\_USER\_ID,

USER\_NAME,

USER\_TYPE,

USER\_START\_DATE,

USER\_END\_DATE,

FND\_USER\_ID,

RESPONSIBILITY\_ID,

RESP\_APPLICATION\_ID,

ATTRIBUTE\_MODULE\_ID,

ATTRIBUTE\_CODE,

ATTR\_MULTIPLE\_VALUES\_FLAG,

ATTRIBUTE\_CATEGORY\_CODE,

MODULE\_CODE,

PROPERTY\_NAME,

PROPERTY\_DESCRIPTION,

FIELD1,

VALUE1,

FIELD2,

VALUE2,

FIELD3,

VALUE3,

FIELD4,

VALUE4,

FIELD5,

VALUE5,

PARENT\_PROPERTY\_ID,

PROP\_START\_DATE,

PROP\_END\_DATE

)

AS

SELECT uauth.user\_attr\_mod\_id,

uauth.property\_id,

uauth.sm\_user\_id,

usr.user\_name,

usr.user\_type,

usr.start\_date user\_start\_date,

usr.end\_date user\_end\_date,

(CASE usr.user\_type WHEN 'A' THEN usr.external\_id ELSE NULL END)

fnd\_user\_id,

(CASE usr.user\_type WHEN 'R' THEN usr.external\_id ELSE NULL END)

responsibility\_id,

(CASE usr.user\_type

WHEN 'R' THEN usr.external\_application\_id

ELSE NULL

END)

resp\_application\_id,

prop.attribute\_module\_id,

prop.attribute\_code,

prop.attr\_multiple\_values\_flag,

prop.attribute\_category\_code,

prop.module\_code,

prop.name property\_name,

prop.description property\_description,

prop.field1,

uauth.value1,

prop.field2,

uauth.value2,

prop.field3,

uauth.value3,

prop.field4,

uauth.value4,

prop.field5,

uauth.value5,

prop.parent\_property\_id,

prop.start\_date prop\_start\_date,

prop.end\_date prop\_end\_date

FROM n\_sm\_properties\_v prop,

n\_security\_mgr\_users usr,

n\_sm\_effective\_user\_auths uauth

WHERE prop.property\_id = uauth.property\_id

AND usr.sm\_user\_id = uauth.sm\_user\_id

AND usr.user\_type <> 'M';

CREATE OR REPLACE FORCE VIEW NOETIX\_VIEWS.N\_SM\_APPS\_USER\_RESP\_SG\_V

(

USER\_ID,

USER\_NAME,

USER\_RESP\_START\_DATE,

USER\_RESP\_END\_DATE,

RESPONSIBILITY\_ID,

APPLICATION\_ID,

RESPONSIBILITY\_NAME,

RESP\_START\_DATE,

RESP\_END\_DATE,

SECURITY\_GROUP\_ID,

SECURITY\_GROUP\_NAME,

RESP\_SEC\_GROUP\_NAME

)

AS

SELECT sec.user\_id,

sec.user\_name,

sec.user\_resp\_start\_date,

sec.user\_resp\_end\_date,

sec.responsibility\_id,

sec.application\_id,

sec.responsibility\_name,

sec.resp\_start\_date,

sec.resp\_end\_date,

sec.security\_group\_id,

sg.security\_group\_name,

sec.responsibility\_name

|| (CASE APPS.XXNAO\_FND\_WRAPPER\_PKG.Are\_Security\_Groups\_Enabled

WHEN 'Y' THEN ':' || sg.security\_group\_name

ELSE TO\_CHAR (NULL)

END)

resp\_sec\_group\_name

FROM n\_fnd\_security\_groups\_vl sg, n\_apps\_security\_info\_vl sec

WHERE SYSDATE BETWEEN sec.user\_resp\_start\_date

AND NVL (sec.user\_resp\_end\_date, SYSDATE)

AND SYSDATE BETWEEN sec.resp\_start\_date

AND NVL (sec.resp\_end\_date, SYSDATE)

AND sg.security\_group\_id(+) = sec.security\_group\_id;

CREATE OR REPLACE FORCE VIEW NOETIX\_VIEWS.N\_SM\_ALL\_EFF\_USER\_AUTHS\_V

(

USER\_ATTR\_MOD\_ID,

PROPERTY\_ID,

SM\_USER\_ID,

ACCESS\_MODE,

USER\_NAME,

USER\_TYPE,

EXTERNAL\_USER\_NAME,

USER\_START\_DATE,

USER\_END\_DATE,

FND\_USER\_ID,

RESPONSIBILITY\_ID,

RESP\_APPLICATION\_ID,

GROUP\_ID,

GROUP\_NAME,

GROUP\_START\_DATE,

GROUP\_END\_DATE,

ATTRIBUTE\_MODULE\_ID,

ATTRIBUTE\_CODE,

ATTR\_MULTIPLE\_VALUES\_FLAG,

ATTRIBUTE\_CATEGORY\_CODE,

MODULE\_CODE,

PROPERTY\_NAME,

PROPERTY\_DESCRIPTION,

FIELD1,

VALUE1,

FIELD2,

VALUE2,

FIELD3,

VALUE3,

FIELD4,

VALUE4,

FIELD5,

VALUE5,

PARENT\_PROPERTY\_ID,

PROP\_START\_DATE,

PROP\_END\_DATE

)

AS

SELECT uauth.user\_attr\_mod\_id,

uauth.property\_id,

uauth.sm\_user\_id,

ua.access\_mode,

usr.user\_name,

usr.user\_type,

usr.external\_user\_name,

usr.start\_date user\_start\_date,

usr.end\_date user\_end\_date,

(CASE usr.user\_type WHEN 'A' THEN usr.external\_id ELSE NULL END)

fnd\_user\_id,

(CASE usr.user\_type WHEN 'R' THEN usr.external\_id ELSE NULL END)

responsibility\_id,

(CASE usr.user\_type

WHEN 'R' THEN usr.external\_application\_id

ELSE NULL

END)

resp\_application\_id,

TO\_NUMBER (NULL) GROUP\_ID,

TO\_CHAR (NULL) group\_name,

TO\_DATE (NULL) group\_start\_date,

TO\_DATE (NULL) group\_end\_date,

prop.attribute\_module\_id,

prop.attribute\_code,

prop.attr\_multiple\_values\_flag,

prop.attribute\_category\_code,

prop.module\_code,

prop.name property\_name,

prop.description property\_description,

prop.field1,

uauth.value1,

prop.field2,

uauth.value2,

prop.field3,

uauth.value3,

prop.field4,

uauth.value4,

prop.field5,

uauth.value5,

prop.parent\_property\_id,

prop.start\_date prop\_start\_date,

prop.end\_date prop\_end\_date

FROM n\_sm\_properties\_v prop,

n\_security\_mgr\_users usr,

n\_sm\_user\_attributes ua,

n\_sm\_effective\_user\_auths uauth

WHERE prop.property\_id = uauth.property\_id

AND usr.sm\_user\_id = uauth.sm\_user\_id

AND ua.user\_attr\_mod\_id = uauth.user\_attr\_mod\_id

AND ua.sm\_user\_id = usr.sm\_user\_id

AND ua.attribute\_module\_id = prop.attribute\_module\_id

AND ua.access\_mode = 'C'

AND usr.user\_type <> 'M'

UNION ALL

SELECT ua.user\_attr\_mod\_id,

ga.property\_id,

ua.sm\_user\_id,

ua.access\_mode access\_mode,

usr.user\_name,

usr.user\_type,

usr.external\_user\_name,

usr.start\_date user\_start\_date,

usr.end\_date user\_end\_date,

(CASE usr.user\_type WHEN 'A' THEN usr.external\_id ELSE NULL END)

fnd\_user\_id,

(CASE usr.user\_type WHEN 'R' THEN usr.external\_id ELSE NULL END)

responsibility\_id,

(CASE usr.user\_type

WHEN 'R' THEN usr.external\_application\_id

ELSE NULL

END)

resp\_application\_id,

ga.GROUP\_ID,

ga.group\_name,

ga.group\_start\_date group\_start\_date,

ga.group\_end\_date group\_end\_date,

ga.prop\_attribute\_module\_id attribute\_module\_id,

ga.prop\_attribute\_code attribute\_code,

ga.attr\_multiple\_values\_flag,

ga.prop\_attribute\_category\_code attribute\_category\_code,

ga.prop\_module\_code module\_code,

ga.prop\_name property\_name,

ga.prop\_description property\_description,

ga.field1,

ga.value1,

ga.field2,

ga.value2,

ga.field3,

ga.value3,

ga.field4,

ga.value4,

ga.field5,

ga.value5,

ga.parent\_property\_id,

ga.prop\_start\_date prop\_start\_date,

ga.prop\_end\_date prop\_end\_date

FROM n\_security\_mgr\_users usr,

n\_sm\_user\_attributes ua,

n\_sm\_eff\_group\_auths\_v ga

WHERE usr.user\_type IN ('A', 'R')

AND ua.sm\_user\_id = usr.sm\_user\_id

AND ga.group\_attribute\_module\_id = ua.attribute\_module\_id

AND ga.group\_access\_mode = 'A'

AND ua.access\_mode = 'A'

AND ga.group\_access\_mode = ua.access\_mode

AND ga.group\_attribute\_code = ga.prop\_attribute\_code

AND ga.group\_user\_type IN ('A', 'R')

AND ( ( usr.user\_type = 'A'

AND ga.group\_external\_id = usr.external\_id

AND ga.group\_external\_application\_id IS NULL)

OR (usr.user\_type = 'R'

AND ga.group\_external\_id = usr.external\_id

AND ga.group\_external\_application\_id =

usr.external\_application\_id))

-- AND NVL(ga.value1,'NULLARG') <> '-9999'

AND ga.group\_attribute\_code = ga.prop\_attribute\_code

AND ga.group\_module\_code = ga.prop\_module\_code

AND ga.group\_attribute\_module\_id = ga.prop\_attribute\_module\_id

AND SYSDATE BETWEEN NVL (ga.group\_start\_date, SYSDATE)

AND NVL (ga.group\_end\_date, SYSDATE + 1)

AND SYSDATE BETWEEN NVL (ga.prop\_start\_date, SYSDATE)

AND NVL (ga.prop\_end\_date, SYSDATE + 1)

UNION ALL

SELECT ua.user\_attr\_mod\_id,

ga.property\_id,

ua.sm\_user\_id,

ua.access\_mode access\_mode,

usr.user\_name,

usr.user\_type,

usr.external\_user\_name,

usr.start\_date user\_start\_date,

usr.end\_date user\_end\_date,

(CASE usr.user\_type WHEN 'A' THEN usr.external\_id ELSE NULL END)

fnd\_user\_id,

(CASE usr.user\_type WHEN 'R' THEN usr.external\_id ELSE NULL END)

responsibility\_id,

(CASE usr.user\_type

WHEN 'R' THEN usr.external\_application\_id

ELSE NULL

END)

resp\_application\_id,

ga.GROUP\_ID,

ga.group\_name,

ga.group\_start\_date group\_start\_date,

ga.group\_end\_date group\_end\_date,

ga.prop\_attribute\_module\_id attribute\_module\_id,

ga.prop\_attribute\_code attribute\_code,

ga.attr\_multiple\_values\_flag,

ga.prop\_attribute\_category\_code attribute\_category\_code,

ga.prop\_module\_code module\_code,

ga.prop\_name property\_name,

ga.prop\_description property\_description,

ga.field1,

ga.value1,

ga.field2,

ga.value2,

ga.field3,

ga.value3,

ga.field4,

ga.value4,

ga.field5,

ga.value5,

ga.parent\_property\_id,

ga.prop\_start\_date prop\_start\_date,

ga.prop\_end\_date prop\_end\_date

FROM n\_security\_mgr\_users usr,

n\_sm\_user\_attributes ua,

n\_sm\_eff\_group\_auths\_v ga

WHERE usr.user\_type <> 'M'

AND ua.sm\_user\_id = usr.sm\_user\_id

AND ga.group\_attribute\_module\_id = ua.attribute\_module\_id

AND ga.group\_access\_mode = 'F'

AND ua.access\_mode = 'F'

AND ga.group\_access\_mode = ua.access\_mode

AND ga.group\_attribute\_code = ga.prop\_attribute\_code

AND ga.group\_user\_type IS NULL

-- AND NVL(ga.value1,'NULLARG') <> '-9999'

AND ga.group\_attribute\_code = ga.prop\_attribute\_code

AND ga.group\_module\_code = ga.prop\_module\_code

AND ga.group\_attribute\_module\_id = ga.prop\_attribute\_module\_id

AND SYSDATE BETWEEN NVL (ga.group\_start\_date, SYSDATE)

AND NVL (ga.group\_end\_date, SYSDATE + 1)

AND SYSDATE BETWEEN NVL (ga.prop\_start\_date, SYSDATE)

AND NVL (ga.prop\_end\_date, SYSDATE + 1);

CREATE OR REPLACE FORCE VIEW NOETIX\_VIEWS.N\_API\_QUERY\_USER\_ROLES\_V

(

USER\_NAME,

ROLE\_NAME

)

AS

SELECT --

qu.user\_name, qur.role\_name

FROM n\_query\_user\_roles qur, n\_query\_users qu

WHERE qur.delete\_flag = 'N'

AND qu.user\_name = qur.user\_name

AND qu.user\_type NOT IN ('A', 'R', 'M')

--

UNION

SELECT rl.user\_name, qur.role\_name

FROM APPS.nv\_nqs\_responsibility\_list rl,

n\_query\_users qu,

n\_query\_user\_roles qur

WHERE qu.user\_type = 'A'

AND qu.user\_name = '#' || rl.user\_id

AND qu.user\_name = qur.user\_name

AND qur.delete\_flag = 'N'

UNION

SELECT rl.user\_name, qur.role\_name

FROM APPS.nv\_nqs\_responsibility\_list rl,

n\_query\_users qu,

n\_query\_user\_roles qur

WHERE qu.user\_type = 'R'

AND qu.user\_name =

'#' || rl.responsibility\_id || '#' || rl.application\_id

AND qu.user\_name = qur.user\_name

AND qur.delete\_flag = 'N'

--

;

CREATE OR REPLACE FORCE VIEW NOETIX\_VIEWS.N\_API\_QUERY\_USERS\_V

(

USER\_NAME,

USER\_TYPE

)

AS

SELECT user\_name, user\_type

FROM n\_query\_users

WHERE delete\_flag = 'N' AND user\_type NOT IN ('A', 'R', 'M')

--

UNION

SELECT rl.user\_name, qu.user\_type

FROM APPS.nv\_nqs\_responsibility\_list rl, n\_query\_users qu

WHERE qu.user\_type = 'A' AND qu.user\_name = '#' || rl.user\_id

UNION

SELECT rl.user\_name, 'A'

FROM APPS.nv\_nqs\_responsibility\_list rl, n\_query\_users qu

WHERE qu.user\_type = 'R'

AND qu.user\_name =

'#' || rl.responsibility\_id || '#' || rl.application\_id

--

;

CREATE OR REPLACE FORCE VIEW NOETIX\_VIEWS.N\_API\_ORACLE\_USERS\_V

(USERNAME, READ\_ONLY\_FLAG, APPS\_USER\_FLAG)

AS

SELECT oracle\_username,

read\_only\_flag,

case when read\_only\_flag in ('O','K','M','U')

then 'Y'

else 'N'

end apps\_user\_flag

-- NOTE: The fnd\_oracle\_userid synonym still exists in the 6.4.1 installer,

-- so there is no need to use the fnd\_oracle\_userid\_s synonym. Makes

-- it compatible with earlier versions.

FROM fnd\_oracle\_userid

--

/\* --

SELECT oprid,

'Y',

case when oprid = 'APPS'

then 'Y'

else 'N'

end apps\_user\_flag

FROM psoprdefn

WHERE acctlock = 0

UNION

SELECT 'APPS',

'Y',

'Y'

FROM dual

-- \*/

;

CREATE OR REPLACE FORCE VIEW NOETIX\_VIEWS.N\_API\_MODULE\_NAMES\_V

(

APPLICATION\_LABEL,

APPLICATION\_SHORT\_NAME,

MODULE\_LONG\_NAME

)

AS

SELECT application\_label,

application\_short\_name,

NVL (

module\_long\_name,

DECODE (application\_label,

'APPS', 'Oracle Applications',

'IFIN', 'Financial Interface',

'NOETIX', 'Noetix Objects',

'PBE', 'Projects - Billing Extension',

'PS', 'General Ledger (Public Sector)',

'XO', 'Cross Operation Extension',

'XXNAO', 'Noetix Administration',

'XXEIS', 'Noetix Administration',

application\_label))

module\_long\_name

FROM n\_api\_appl\_module\_names\_v;

CREATE OR REPLACE FORCE VIEW NOETIX\_VIEWS.N\_API\_BI\_USERS\_V

(

SM\_USER\_ID,

USER\_NAME,

MAPPED\_USER\_NAME,

MAPPED\_SERVER\_NAME,

SM\_MAPPED\_USER\_ID,

SM\_EXTERNAL\_SERVER\_ID,

EBS\_USER\_ID

)

AS

SELECT TO\_NUMBER (NULL),

TO\_CHAR (NULL),

TO\_CHAR (NULL),

TO\_CHAR (NULL),

TO\_NUMBER (NULL),

TO\_NUMBER (NULL),

TO\_CHAR (NULL)

FROM DUAL

WHERE 1 = 0

--

UNION ALL

SELECT m.sm\_user\_id,

m.user\_name,

a.external\_user\_name AS mapped\_user\_name,

s.user\_server\_name AS mapped\_server\_name,

a.sm\_user\_id AS sm\_mapped\_user\_id,

s.external\_server\_id external\_server\_id,

a.external\_id AS ebs\_user\_id

FROM n\_security\_mgr\_users m

JOIN n\_security\_mgr\_users a

ON m.effective\_sm\_user\_id = a.sm\_user\_id

JOIN n\_sm\_external\_servers s

ON s.external\_server\_id = m.external\_server\_id

WHERE m.user\_type = 'M' AND a.user\_type = 'A'