Oracle® Healthcare Translational Research

Administrator's Guide

Release 3.2

E86015-04

September 2017

Oracle Healthcare Translational Research Administrator's Guide, Release 3.2

E86015-04

Contents

Preface

This guide describes how to perform administrative tasks in Oracle Healthcare Translational Research (OHTR).

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Finding More Information

Oracle Help Center

The latest user documentation for Oracle Health Sciences products is available at http://docs.oracle.com/en/industries/health-sciences/.

My Oracle Support

The latest release notes, patches and white papers are on My Oracle Support (MOS) at https://support.oracle.com. For help with using MOS, see https://docs.oracle.com/cd/E74665_01/MOSHP/toc.htm.

Related Documents

Oracle Business Intelligence Enterprise Edition Documentation

OBIEE documentation is available at http://docs.oracle.com/middleware/12212/biee/index.html.

Oracle Healthcare Foundation

OHF documentation is available at https://docs.oracle.com/en/industries/health-sciences/oracle-healthcare-foundation/index.html

1

Managing Access to Patient or Subject Records and PII

Use data access policies to control users' access to patient and subject data in several ways:

- **Control access to Personally Identifiable Information (PII) attributes**. Create data access policies that specify which Personally Identifiable Information (PII) attributes are visible for subjects in a particular study or patients in a particular patient group, to users assigned to the configuration. PII attributes that are not visible are obfuscated (masking values are displayed).

  This functionality is always enabled so that anyone needing access to PII data must be assigned to a data access policy that grants the required access.

- **Control access to non-PII subject or patient information** by enabling row-level filtering. This setting applies across all studies and patient groups.
  - If disabled (the default state), users can see all non-PII subject or patient data in any study or patient group.
  - If enabled, only users assigned to a data access policy that allows access to a study or patient group can see any non-PII subject or patient data in the study or patient group.
- **Allow access to all data** by assigning a user to a global configuration that allows access to all subject and patient data, including PII data, in all studies and patient groups.

If a user has access to the same study or patient group through multiple data access policies, if any one data access policy permits access to a particular subject or patient's data, it is visible to the user.

- Creating a Data Access Policy
- Assigning a User to a Data Access Policy
- Deactivating a Data Access Policy
- Limiting Access to non-PII Data in Patient and Subject Records
- Granting Access to All Subject and Patient Data
- Attribute Groups
- Sample Use Cases

In previous versions of OHTR, data access policies were called *VPD configurations*.

1.1 Creating a Data Access Policy

Specify a study or patient group and a set of personally identifiable information (PII) that should be visible to certain users. All users assigned to the data access policy have access to the specified PII data for the subjects or patients in the specified study or patient group. If some users should have access to a different set of PII data for the same study or patient group, create a different data access policy for them.

1. On the database server, log in to SQL*Plus as CDM.
2. Run VPD_UTIL.ADD_VPD_CONFIG, entering values as follows:
   - A unique name for the data access policy.
   - The name of one study or patient group whose data will be accessible through the data access policy.
   - A description of the data access policy.
   - A value for every attribute group parameter containing either `subj` or `pt`, depending on whether you specified a study or a patient group.

     A value of `Y` allows access. `N` prevents access. For a description of each attribute group, see:

     Table 1–1, " Personally Identifiable Attribute Groups for Subjects"

     Table 1–2, " Personally Identifiable Attribute Groups for Patients"
   - (Optional) An expiration date for the data access policy in the format `YYYY-MM-DD`. On this date the data access policy will be automatically deactivated and any users assigned to it will no longer be able to access subject/patient data through it.

Example 1 - Data Access Policy with Some PII Access to 'Study B' Subjects

```
BEGIN
VPD_UTIL.ADD_VPD_CONFIG    (I_CONFIG_NAME  =>'STUDY_B_SUBJECTS_FULL_PII' ,
I_SUBJ_ADD     =>'N'                                    ,
I_SUBJ_CONSENT =>'Y'                                    ,
I_SUBJ_DATE    =>'Y'                                    ,
I_SUBJ_DX      =>'Y'                                    ,
I_SUBJ_ENC     =>'Y'                                    ,
I_SUBJ_HIST    =>'Y'                                    ,
I_SUBJ_ID      =>'Y'                                    ,
I_SUBJ_MED     =>'Y'                                    ,
I_SUBJ_NAME    =>'N'                                    ,
I_SUBJ_OBSV    =>'Y'                                    ,
I_SUBJ_PROC    =>'Y'                                    ,
I_SUBJ_SPEC   =>'Y'                                    ,
I_DESCRIPTION  =>'data access policy with access to all Study B subjects with all
PII attributes except subject name and address',
I_SUBJECT_STUDY=>'Study B'
);
END;
/
```

Example 2 - Data Access Policy with Full PII Access to GROUP1 Patients

```
BEGIN
VPD_UTIL.ADD_VPD_CONFIG    (I_CONFIG_NAME  =>'GROUP1_PATIENTS_ALL_PII' ,
I_PT_ADD     =>'Y'                                    ,
I_PT_CONSENT =>'Y'                                    ,
I_PT_DATE    =>'Y'                                    ,
I_PT_DX      =>'Y'                                    ,
I_PT_ENC     =>'Y'                                    ,
I_PT_HIST    =>'Y'                                    ,
I_PT_ID      =>'Y'                                    ,
I_PT_MED     =>'Y'                                    ,
I_PT_NAME    =>'Y'                                    ,
I_PT_OBSV    =>'Y'                                    ,
I_PT_PROC    =>'Y'                                    ,
I_PT_SPEC =>'Y'                                    ,
I_DESCRIPTION  =>'data access policy with access to GROUP1 patients for all PII
attributes',
I_PATIENT_GROUP  =>'GROUP1',
I_EXPIRATION_DATE       => date '2025-12-31'
);
END;
/
```

1.2 Assigning a User to a Data Access Policy

Assign users to a data access policy to give them permission to see the specified PII for the specified subjects or patients. You can assign either WebLogic user accounts or database user accounts, one account at a time.

1. On the database server, log in to SQL*Plus as CDM.
2. Run VPD_UTIL.ADD_CONFIG_USER, entering values as follows:
   - The data access policy name.
   - The user's user name.
   - (Optional) An expiration date for the user assignment in the format `YYYY-MM-DD`. On this date the user will be automatically deassigned and will no longer be able to access subject/patient data through the data access policy.

   For example:

   ```
   BEGIN
   VPD_UTIL.ADD_CONFIG_USER
      (I_EXISTING_CONFIG_NAME  =>'PATIENT_GROUP_1_ALL_ATTRIBUTES' ,
      I_USER_NAME  => 'TESTER' ,
      I_EXPIRATION_DATE       => DATE '2025-12-31'   );
    END;/
   ```

1.3 Deactivating a Data Access Policy

Deactivating a data access policy removes the specified data access from all the users assigned to it, though users may have access through a different data access policy.

1. On the database server, log in to SQL*Plus as CDM.
2. Run stored procedure VPD_UTIL.INACTIVATE_CONFIG, entering the data access policy name for I_EXISTING_CONFIG_NAME:

   ```
   exec VPD_UTIL.INACTIVATE_CONFIG;
   ```

1.4 Limiting Access to non-PII Data in Patient and Subject Records

By default, any application user is permitted to access non-PII records for all patients and subjects. If required, the row-level filtering mode can be turned on at the system level, which limits user access to only a subset of patients and/or subjects. If this optional mode is turned on, only users explicitly assigned to a data access policy are granted access to the patients or subjects associated with the policy. A user can be assigned to any number of policies.

1. On the database server, log in to SQL*Plus as CDM.
2. Run stored procedure VPD_UTIL.ENABLE_ROW_FILTER_POLICIES:
   - To require that users must be assigned to a data access policy to see any subject or patient data for a particular study or patient group, enter a value of `1`:

     ```
     exec vpd_util.enable_row_filter_policies(1)
     ```

   - To enable all users to see non-PII data for any subject or patient, enter a value of `0`. This is the default value.

     ```
     exec vpd_util.enable_row_filter_policies(0)
     ```

1.5 Granting Access to All Subject and Patient Data

A global data access policy permits access to all patients and subjects and all their PII attribute values. Users who are assigned to this data access policy do not need to be assigned to any other data access policy, even if row filtering is on. Its ID value is `1`.

1. On the database server, log in to SQL*Plus as CDM.
2. Run stored procedure VPD_UTILADD_CONFIG_USER entering values as follows:
   - The configuration ID set to `1`.
   - The user’s user name.
   - (Optional) An expiration date for the user assignment in the format `YYYY-MM-DD`. On this date the user will be automatically deassigned and will no longer be able to access subject/patient data through the data access policy.

   For example:

   ```
   BEGIN
   VPD_UTIL.ADD_CONFIG_USER
     (I_EXISTING_CONFIG_ID  =>1 ,
      I_USER_NAME  => 'JSMITH' ,
      I_EXPIRATION_DATE       => DATE '2025-12-31'   );
    END;
   /
   ```

1.6 Attribute Groups

PII attributes are combined into PII attribute groups. data access policies grant data access to the group, not to individual attributes. See:

- Table 1–1, " Personally Identifiable Attribute Groups for Subjects"
- Table 1–2, " Personally Identifiable Attribute Groups for Patients"

Table 1–1 Personally Identifiable Attribute Groups for Subjects

Subject Attribute Groups	Description	Table	Column(s)	API Procedure Input Parameter
SUBJ_ADD	Subject Address	W_EHA_SUBJECT_D	CITY, POSTAL_CODE, STREET_ADDRESS_1, STREET_ADDRESS_2, STREET_ADDRESS_3	I_SUBJ_ADD
SUBJ_CONSENT	Subject Consent Dates	W_EHA_ENC_ PATIENT_H	CONSENT_START_DT, CONSENT_END_DT	I_SUBJ_CONSENT
SUBJ_DATE	Subject Lifecycle Dates	W_EHA_SUBJECT_D	DOB, DECEASED_DT	I_SUBJ_DATE
SUBJ_DX	Subject Diagnosis Dates	W_EHA_DX_ SUBJECT_H	DIAGNOSIS_ONSET_DT, DIAGNOSIS_REPORTED_ DT, DIAGNOSIS_END_ DT, AGE_AT_FIRST_ ONSET	I_SUBJ_DX
SUBJ_ENC	Subject Encounter Dates	W_EHA_ENC_ SUBJECT_H	ENCOUNTER_START_ DT, ENCOUNTER_END_ DT	I_SUBJ_ENC
SUBJ_HIST	Subject History	W_EHA_SBJ_ HISTORY_SBJ_H	SUBJECT_HISTORY_ START_DT, SUBJECT_ HISTORY_END_DT	I_SUBJ_HIST
SUBJ_ID	Subject Identifier	W_EHA_SUBJECT_D	SUBJECT_IDENTIFIER	I_SUBJ_ID
SUBJ_MED	Subject Medication Dates	W_EHA_SUBADMN_ SUBJECT_H	SUBADMN_START_DT, SUBADMN_END_DT	I_SUBJ_MED
SUBJ_NAME	Subject Name	W_EHA_SUBJECT_D	FIRST_NAME, MIDDLE_ NAME, LAST_NAME	I_SUBJ_NAME
SUBJ_OBSV	Subject Observation Dates	W_EHA_OBSV_ SUBJECT_H	OBSV_DT	I_SUBJ_OBSV
SUBJ_PROC	Subject Procedure Dates	W_EHA_PROC_ SUBJECT_H	PROCEDURE_START_DT, PROCEDURE_END_DT	I_SUBJ_PROC
SUBJ_SPEC	Subject Specimen Identifier and Collection Date	W_EHA_SPECIMEN_ SUBJECT_H	SPECIMEN_ COLLECTION_DT, SPECIMEN_NUMBER	I_SUBJ_SPEC

Table 1–2 Personally Identifiable Attribute Groups for Patients

Patient Attribute Groups	Description	Table	Column(s)	API Procedure Input Parameter
PT_ADD	Patient Address	W_EHA_ RESEARCH_ PATIENT_D	CITY, POSTAL_CODE, STREET_ADDRESS_1, STREET_ADDRESS_2, STREET_ADDRESS_3	I_PT_ADD
PT_CONSENT	Patient Consent Dates	W_EHA_ CONSENT_ PATIENT_H	CONSENT_START_DT, CONSENT_END_DT	I_PT_CONSENT
PT_DATE	Patient Lifecycle Dates	W_EHA_ RESEARCH_ PATIENT_D	DOB, DECEASED_DT	I_PT_DATE

Table 1–2 (Cont.) Personally Identifiable Attribute Groups for Patients

Patient Attribute Groups	Description	Table	Column(s)	API Procedure Input Parameter
PT_DX	Patient Diagnosis Dates	W_EHA_DX_ PATIENT_H	DIAGNOSIS_ONSET_DT, DIAGNOSIS_ REPORTED_DT, DIAGNOSIS_END_DT, AGE_AT_FIRST_ONSET	I_PT_DX
PT_ENC	Patient Encounter Dates	W_EHA_ENC_ PATIENT_H	ENCOUNTER_START_ DT, ENCOUNTER_END_ DT	I_PT_ENC
PT_HIST	Patient History	W_EHA_PT_ HISTORY_PT_H	PATIENT_HISTORY_ START_DT, PATIENT_ HISTORY_END_DT	I_PT_HIST
PT_ID	Patient Identifier	W_EHA_ RESEARCH_ PATIENT_D	PATIENT_IDENTIFIER	I_PT_ID
PT_MED	Patient Medication Dates	W_EHA_ SUBADMN_ PATIENT_H	SUBADMN_START_DT, SUBADMN_END_DT	I_PT_MED
PT_NAME	Patient Name	W_EHA_ RESEARCH_ PATIENT_D	FIRST_NAME, MIDDLE_ NAME, LAST_NAME	I_PT_NAME
PT_OBSV	Patient Observation Dates	W_EHA_OBSV_ PATIENT_H	OBSV_DT	I_PT_OBSV
PT_PROC	Patient Procedure Dates	W_EHA_PROC_ PATIENT_H	PROCEDURE_START_DT, PROCEDURE_END_DT	I_PT_PROC
PT_SPEC	Patient Specimen Identifier and Collection Date	W_EHA_ SPECIMEN_ PATIENT_H	SPECIMEN_ COLLECTION_DT, SPECIMEN_NUMBER	I_PT_SPEC

1.7 Sample Use Cases

Scenario 1: Dr. Smith needs to view all patient and subject data with de-identified PII. She works in an environment where row-level filtering is disabled, meaning assignment to a data access policy is not required.

There is no need to explicitly assign a data access policy to Dr. Smith. In her environment, any user has access to non-PII data for all patients and subjects.

Scenario 2: Dr. Chen needs to see patient data in Patient Group 1, including all PII values except patient name and address. She works in an environment where access to patient and subject records is controlled.

1. Create a data access policy for the patient group that Dr. Chen is has access to view. Set all patient attribute groups in the policy except Patient name and address to `Y`.

```
BEGIN
VPD_UTIL.ADD_VPD_CONFIG    (I_CONFIG_NAME  =>'GROUP1_PATIENTS_ALL_PII_EXCEPT_
NAME_ADDRESS' ,
I_PT_ADD     =>'N'                                    ,
I_PT_CONSENT =>'Y'                                    ,
I_PT_DATE    =>'Y'                                    ,
I_PT_DX      =>'Y'                                    ,
I_PT_ENC     =>'Y'                                    ,
I_PT_HIST    =>'Y'                                    ,
I_PT_ID      =>'Y'                                    ,
I_PT_MED     =>'Y'                                    ,
I_PT_NAME    =>'N'                                    ,
I_PT_OBSV    =>'Y'                                    ,
I_PT_PROC    =>'Y'                                    ,
I_PT_SPEC =>'Y'                                    ,
I_DESCRIPTION  =>'Configuration with access to GROUP1 patients with PII
attributes except Name and Address',
I_PATIENT_GROUP  =>'GROUP1',
I_EXPIRATION_DATE       => date '2025-12-31'
);
END;
/
```

2. Assign Dr. Chen (jchen12) to the above data access policy.

```
BEGIN
VPD_UTIL.ADD_CONFIG_USER
  (I_EXISTING_CONFIG_NAME  => 'GROUP1_PATIENTS_ALL_PII_EXCEPT_NAME_ADDRESS'  ,
   I_USER_NAME             => 'jchen12',
   I_EXPIRATION_DATE       => date '2025-12-31');
END;
/
```

Scenario 3: Dr. Gupta is authorized to see all patient and subject data, including identifiable data.

Assign the preconfigured global data access policy to Dr. Gupta.

```
BEGIN
VPD_UTIL.ADD_CONFIG_USER
  (I_EXISTING_CONFIG_ID    => 1,
   I_USER_NAME             => 'kgupta',
   I_EXPIRATION_DATE       => date '2025-12-31');
END;
/
```

Scenario 4: Dr. Black needs to see de-identified data in STUDY A and identified data in STUDY B.

1. Create a data access policy that grants access to de-identified PII data on subjects from STUDY A.

> **Note:** This step is optional when row-level filtering is disabled

```
BEGIN
VPD_UTIL.ADD_VPD_CONFIG    (I_CONFIG_NAME  =>'STUDY_A_SUBJECTS_NO_PII' ,
  I_SUBJ_ADD     =>'N'                                     ,
I_SUBJ_CONSENT =>'N'                                    ,
I_SUBJ_DATE    =>'N'                                    ,
I_SUBJ_DX      =>'N'                                    ,
I_SUBJ_ENC     =>'N'                                    ,
I_SUBJ_HIST    =>'N'                                    ,
I_SUBJ_ID      =>'N'                                    ,
I_SUBJ_MED     =>'N'                                    ,
I_SUBJ_NAME    =>'N'                                    ,
I_SUBJ_OBSV    =>'N'                                    ,
I_SUBJ_PROC    =>'N'                                    ,
I_SUBJ_SPEC   =>'N'                                    ,
  I_DESCRIPTION  =>'Configuration with access to Study A subjects with no PII
attribute values',
  I_SUBJECT_STUDY=>'Study A'
);
END;
/
```

2. Create a data access policy that grants access to subjects from STUDY B and their identifiable attribute values:

```
BEGIN
VPD_UTIL.ADD_VPD_CONFIG    (I_CONFIG_NAME  =>'STUDY_B_SUBJECTS_FULL_PII' ,
I_SUBJ_ADD     =>'Y'                                    ,
I_SUBJ_CONSENT =>'Y'                                    ,
I_SUBJ_DATE    =>'Y'                                    ,
I_SUBJ_DX      =>'Y'                                    ,
I_SUBJ_ENC     =>'Y'                                    ,
I_SUBJ_HIST    =>'Y'                                    ,
I_SUBJ_ID      =>'Y'                                    ,
I_SUBJ_MED     =>'Y'                                    ,
I_SUBJ_NAME    =>'Y'                                    ,
I_SUBJ_OBSV    =>'Y'                                    ,
I_SUBJ_PROC    =>'Y'                                    ,
I_SUBJ_SPEC   =>'Y'                                    ,
I_DESCRIPTION  =>'Configuration with access to Study B subjects with all PII
attribute values',
I_SUBJECT_STUDY=>'Study B'
);
END;
/
```

3. Assign Dr. Black to these data access policies:

> **Note:** Assigning `STUDY_A_SUBJECTS_NO_PII` configuration is optional if row-level filtering is disabled.

```
BEGIN
VPD_UTIL.ADD_CONFIG_USER
  (I_EXISTING_CONFIG_NAME  => 'STUDY_A_SUBJECTS_NO_PII',
   I_USER_NAME             => 'sblack',
   I_EXPIRATION_DATE       => date '2025-12-31');
END;
/

BEGIN
VPD_UTIL.ADD_CONFIG_USER
  (I_EXISTING_CONFIG_NAME  => 'STUDY_B_SUBJECTS_FULL_PII',
   I_USER_NAME             => 'sblack',
   I_EXPIRATION_DATE       => date '2025-12-31');
END;
/
```

2

Managing Access to Specimen Aliases

A laboratory that receives a specimen for processing may assign a barcode and use it for specimen identification purposes. In the CDM schema, these barcode identifiers are tracked as specimen aliases in addition to the primary lab specimen identifier (SPECIMEN_NUMBER/SPECIMEN_VENDOR_NUMBER).

The permissions to see specimen aliases are calculated based on the service provider (lab) that issued the alias and study or patient group that includes the specimen donor (patient or subject).

- Authorizing Specimen Alias Access
- Revoking Specimen Alias Access for a User

2.1 Authorizing Specimen Alias Access

For each user who needs access to specimen aliases, specify either a specific service provider (such as a lab) or all providers and either a specific study or patient group or all studies or all patient groups. If a user needs access to specimen aliases from more than one provider but not all, or to specimen aliases used in more than one study or patient group but not all, run this procedure once for each combination required.

1. On the database server, log in to SQL*Plus as CDM.
2. Run stored procedure VPD_UTIL.GRANT_SVCPRV_USER, entering values as follows:
   - The user's user name.
   - Service provider scope. Set one of the following parameters:
     - I_ISSNG_SVCPRV_ID. To limit access to aliases used by a single service provider, enter the service provider's ID.
     - I_ANY_ISSNG_SVCPRVS. To allow access to aliases created by any service provides, set this parameter s to `1`.
   - Subject and/or patient scope. Set one of the following parameters, or set one study parameter and one patient group parameter:
     - I_STUDY_NAME. To allow access to aliases used in a single study, enter the name of the study.
     - I_ANY_STUDY. To allow access to aliases used in all studies, set this parameter to `1`.
     - I_PT_GROUP_NAME. To allow access to aliases used for a single patient group, enter the name of the patient group.

  - I_ANY_PT_GROUP. To allow access to aliases used in all patient groups, set this parameter to `1`.

- (Optional) An expiration date for the user assignment. On this date the privileges given to the user will be automatically revoked. The date can be formatted in any valid date-type expression. For example:

```
 sysdate+x_days
date '2020-12-31'
trunc(sysdate +1217
to_date('2020-12-31','YYYY-MM-DD')
```

Example 1 - Access to a single provider's aliases for a single patient group

Authorize a user to access aliases for a specific service provider in the context of a single patient group.

```
begin
 vpd_util.grant_svcprv_user (
  i_user_name =>' LABUSER1',
  i_pt_group_name =>'GROUP_1',
  i_issng_svcprv_id =>'SVCPRV1',
  i_expiration_date =>sysdate+100
) ;
end;
/
```

Example 2 - Access to a single provider's aliases for a single study

Authorize a user to access aliases for a specific service provider in the context of a single study.

```
begin
 vpd_util.grant_svcprv_user (
  i_user_name =>' LABUSER2',
  i_study_name =>'STUDY3',
  i_issng_svcprv_id =>'18_SVCPRV_NBR',
  i_expiration_date =>sysdate+100
) ;
end;
/
```

Example 3 - Access to any provider's aliases for any study or patient group

Authorize a user to access any specimen alias in the context of any patient group or study.

```
begin
 vpd_util.grant_svcprv_user (
  i_user_name =>' LABSUPEVISOR1',
  i_any_study =>1,
  i_any_pt_group =>1,
  i_any_issng_svcprvs =>1,
  i_expiration_date =>sysdate+100
) ;
end;
/
```

2.2 Revoking Specimen Alias Access for a User

To revoke specimen alias access from a user, use numeric identifiers (row_wids) of the specific study or patient group.

1. On the database server, log in to SQL*Plus as CDM.
2. Run stored procedure VPD_UTILrevoke_svcprv_user, entering values as follows:
   - The user's user name.
   - Service provider scope. Set one of the following parameters:
     - I_ISSNG_SVCPRV_ID. To revoke access from aliases used by a single service provider, enter the service provider's ID.
     - I_ANY_ISSNG_SVCPRVS. To revoke access from aliases created by any service provides, set this parameter s to `1`.
   - Subject and/or patient scope. Set one of the following parameters, or set one study parameter and one patient group parameter:
     - I_STUDY_NAME. To revoke access from aliases used in a single study, enter the name of the study.
     - I_ANY_STUDY. To revoke access from aliases used in all studies, set this parameter to `1`.
     - I_PT_GROUP_NAME. To revoke access from aliases used for a single patient group, enter the name of the patient group.
     - I_ANY_PT_GROUP. To revoke access from aliases used in all patient groups, set this parameter to `1`.

Note: For the purpose of revoking access, the parameters for *any study*, *any pt group*, and *any provider* do not include any separately granted named study, patient, or provider scope. You must revoke these separately if needed.

Example 1 - Revoke access to one study/provider combination

Revoke access to study_wid=99 and Service Provider (row_wid=105) from user TSTUSER1.

```
begin
vpd_util.revoke_svcprv_user (
    i_user_name =>'TSTUSER1',
  i_study_wid =>99,
  i_issng_svcprv_wid =>105
) ;
end;
/
```

Example 2 - Revoke access to one patient group/provider combination

Revoke access to patient_group_wid=10 and Service Provider (row_wid=105) from user TSTUSER2.

```
begin
vpd_util.revoke_svcprv_user (
    i_user_name =>'TSTUSER2',
i_pt_group_wid =>10,
  i_issng_svcprv_wid =>105
) ;
end;
/
```

Example 3 - Revoke access to any study, patient group, and provider granted using parameter I_ANY_STUDY, I_ANY_PT_GROUP, and I_ANY_ISSNG_SVCPRVS

Revoke access from a user to any study or patient group or service provider that was granted through an "any" parameter.

```
begin
vpd_util.revoke_svcprv_user (
    i_user_name =>'TSTUSER3',
  i_any_study =>1,
  i_any_pt_group =>1,
  i_any_issng_svcprvs =>1
) ;
end;
/
```

3
Optimizing Query Engine Performance

To optimize the performance of the query engine, refresh statistics by running stored procedure REFRESH_STATS_TAB.

Execute this procedure:

- After the initial CDM data load.
- After any significant changes in data volumes or distribution in the CDM schema.
- Periodically.

To execute the procedure:

1. On the database server, log in to SQL*Plus as CDM.
2. Execute the following commands:

```
set serveroutput on
set echo on
spool REFRESH_STATS_TAB
execute REFRESH_STATS_TAB
spool off
exit
```