

Appendix I

Oracle Designer Report – Database Trigger

Oracle Designer

Report : DATABASE TRIGGER
Filename : s:\users\rpm\dbtrpml.pdf
Run by THIMONAS
Report Date : 10 February 2004
Total Pages : 123

Parameter Values

Workarea : GLOBAL SHARED WORKAREA
Container : Real Property Management
Container Version :
Recurse Sub-Containers : N
Table/View : %
Trigger Name : %

Database Triggers Created

On/After :
On/Before : 10 February 2004
and

Database Triggers Changed

On/After :
On/Before : 10 February 2004

Container: Real Property Management

Version :

Table/View Name :II_LU_LU_LINKS (table)

Trigger Name :	II_TD_LULULNK_TRG	Create?	Yes
PL/SQL Module			
Name :	II_TD_LULULNK_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	No
		Delete?	Yes
New Alias	NEW	Update?	No
Old Alias	: OLD		

Trigger Logic

```

DECLARE
BEGIN
  DECLARE
    vFromLUCN ii_lu_lu_links.from_lu_cn%TYPE := null;
    vToLUCN   ii_lu_lu_links.to_lu_cn%TYPE   := null;
    cursor cFrom is
      select obj_name
      from land_units
      where cn = vFromLUCN;
    cursor cTo is
      select obj_name
      from land_units
      where cn = vToLUCN;
    vFromObjName land_units.obj_name%TYPE := null;
    vToObjName   land_units.obj_name%TYPE := null;
  BEGIN
    -- Check role assignment for Wilderness-Geolocator deletes.
    IF not ii_chk_role_assign('INFRA_SVC_MGR') THEN
      vFromLUCN := :old.from_lu_cn;
      vToLUCN := :old.to_lu_cn;
      open cFrom;
      fetch cFrom into vFromObjName;
      close cFrom;
      open cTo;
      fetch cTo into vToObjName;
      close cTo;
      IF vFromObjName = 'WILDERNESS' THEN
        IF vToObjName in
          ('STATE','COUNTY','CONGRESSIONAL_DISTRICT',
           'PROCLAIMED_UNIT') then
          raise_application_error(-20302,'Insufficient
privileges to '||
                                'delete a Wilderness-'||
                                initcap(replace(vToObjName,'_',' '))||'
link');
        END IF;
      ELSIF vFromObjName in
        ('STATE','COUNTY','CONGRESSIONAL_DISTRICT',
         'PROCLAIMED_UNIT') then
        IF vToObjName = 'WILDERNESS' then
          raise_application_error(-20302,'Insufficient
privileges to '||

```

Oracle Designer Report : cktrigger

Container: Real Property Management **Version :**

Table/View Name :II_LU_LU_LINKS (table)

```
                                'delete a Wilderness-'||
                                initcap(replace(vFromObjName, '_ ', ' '))||'
link');
                                END IF;
                                END IF;
                                END IF;
                                END;
                                END;
```

Container: Real Property Management Version :

Table/View Name :II_LU_CONT_LINKS (table)

Trigger Name :	II_TD_LUCONT_TRG	Create?	Yes
PL/SQL Module Name :	II_TD_LUCONT_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	No
		Delete?	Yes
New Alias	NEW	Update?	No
Old Alias	OLD		

Trigger Logic

```

DECLARE
BEGIN
  DECLARE
    vLUCN ii_lu_cont_links.lu_cn%TYPE := null;
    cursor cWild is
      select obj_name
      from land_units
      where cn = vLUCN;
    vLUObjName land_units.obj_name%TYPE := null;
  BEGIN
    IF not ii_chk_role_assign('INFRA_SVC_MGR') THEN
      vLUCN := :old.lu_cn;
      open cWild;
      fetch cWild into vLUObjName;
      close cWild;
      IF vLUObjName = 'WILDERNESS' THEN
        IF :old.link_type name = 'ADMINISTRATIVE UNIT' THEN
          raise_application_error(-20301,'Insufficient privileges
to '||
                                'delete a Wilderness-Administrative Unit link');
        END IF;
      END IF;
    END IF;
  END IF;
END;
END;
```

Container: Real Property Management

Version :

Table/View Name :LAND_UNITS (table)

Trigger Name : II_LU_ADR_TRG Create? Yes

PL/SQL Module

Name : II_LU_ADR_TRG Enabled? Yes

PL/SQL File :

Purpose : Keep deleting conciled with InfraCentral

Trigger Time: AFTER Level: ROW Insert? No

Delete? Yes

New Alias Update? No

Old Alias :

Trigger Logic

```
DECLARE
    vThisInstanceId ii_db_instances.instance_id%type;
BEGIN
    SELECT instance_id INTO vThisInstanceId
    FROM this_db_instance;

    ii_rpa_remote.deleteLandUnit@ic_remote(
        vThisInstanceId
        ,:old.cn
        ,USER
        ,SYSDATE);
END;
```

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

Trigger Name :	II_TD_LU_TRG	Create?	Yes
PL/SQL Module			
Name :	II_TD_LU_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	No
		Delete?	Yes
New Alias	NEW	Update?	No
Old Alias	: OLD		

Trigger Logic

```

DECLARE
    vThisInstanceId ii_db_instances.instance_id%type;
BEGIN
    SELECT instance_id INTO vThisInstanceId
    FROM this_db_instance;

    ii_rpa_remote.deleteLandUnit@ic_remote(
        vThisInstanceId
        ,:old.cn
        ,USER
        ,SYSDATE);
END;

DECLARE
BEGIN
    BEGIN
        IF not ii_chk_role_assign('INFRA SVC MGR') and
            ii_cent_mgd_lu_obj(:old.obj_name) then
            raise_application_error(-20301,'Insufficient privileges to
'||
            'delete '||initcap(replace(:old.obj_name,'_',' '))||'
Land Units');
        END IF;
        --
        -- Added by RAdams on 6/24/2002 for v5.0:
        --
        -- Record old/new ID and CN in PL/SQL table for purposes of
        -- writing row to range permits audit table
        (ii_range_pmt_li_audit) when the deleted
        -- Land Unit ID is an RMU, State, Proclaimed Unit, or County
        referenced in a
        -- grazing permit. This action takes place in the
        statement-level
        -- after-update/delete trigger that first queries the range
        permit line items table
        -- to check if this land unit is an RMU, State, Proclaimed
        Unit, or County referenced in
        -- a grazing permit. If this is the case, a re-query on the
        land_units table is necessary
        -- to get the remaining RMU, State, Proclaimed Unit, and
        County IDs to be inserted
        -- into the audit table. To avoid the "mutating table"

```

Oracle Designer Report : cktrigger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

error, the re-query of the land_units
--      table needed to be moved to a statement-level after
update/delete trigger, passing "old" and "new"
--      values in the package PL/SQL table referenced in the code
below.
--
      if nvl(:new.id,'@') != nvl(:old.id,'@') then
        ii_range_permits_audit.vNumEntries :=
ii_range_permits_audit.vNumEntries + 1;
ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).new_id := :new.id;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).old_id := :old.id;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).new_cn := :new.cn;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).old_cn := :old.cn;
      end if;
--
      END;
END;
```

Container: Real Property Management Version :

Table/View Name :II_CONTACTS (table)

Trigger Name : II_CONT_RBD_TRG Create? Yes

PL/SQL Module

Name : II_CONT_RBD_TRG Enabled? Yes

PL/SQL File :

Purpose : Row-level before-delete trigger

Trigger Time: BEFORE Level: ROW Insert? No

Delete? Yes

New Alias Update? No

Old Alias :

Trigger Logic

```

Begin
--
-- Added by RAdams on 6/24/2002 for v5.0:
--
-- Record old/new ID and CN in PL/SQL table for purposes of
-- writing row to range permits audit table
(ii_range_permit_audits) when the deleted
-- Contact ID is a permittee or managing organization for a
-- grazing permit. This action takes place in the statement-level
-- after-update/delete trigger that first queries the range
permits view
-- to check if this contact is a permittee or managing
organization for
-- a grazing permit. If this is the case, a re-query on the
ii_contacts table is necessary
-- to get the remaining Permittee or Managing Org ID to be
inserted
-- into the audit table. To avoid the "mutating table" error, the
re-query of the ii_contacts
-- table needed to be moved to a statement-level after
update/delete trigger, passing "old" and "new"
-- values in the package PL/SQL table referenced in the code
below.
--
if nvl(:new.id,'@') != nvl(:old.id,'@') then
    ii_range_permits_audit.vNumEntries :=
ii_range_permits_audit.vNumEntries + 1;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ne
w_id := :new.id;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ol
d_id := :old.id;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ne
w_cn := :new.cn;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ol
d_cn := :old.cn;
end if;
--
End;
```

Oracle Designer Report : cktrgger

Container: Real Property Management

Version :

Table/View Name :II_FEATURES (table)

Trigger Name : II_FEA_ADR_TRG Create? Yes

PL/SQL Module

Name : II_FEA_ADR_TRG Enabled? Yes

PL/SQL File :

Purpose : Keep deleting conciled with InfraCentral.

Trigger Time: AFTER Level: ROW Insert? No

Delete? Yes

New Alias Update? No

Old Alias :

Trigger Logic

```
DECLARE
  vThisInstanceId ii_db_instances.instance_id%type;
BEGIN
  SELECT instance_id INTO vThisInstanceId
  FROM this_db_instance;

  ii_rpa_remote.deleteFeature@ic_remote(
    vThisInstanceId
    ,:old.cn
    ,USER
    ,SYSDATE);
END;
```

Container: Real Property Management Version :

Table/View Name :II_GSA_USAGE_CODES (table)

Trigger Name : II_GSA_USAGE_CODE Create? Yes

PL/SQL Module

Name : II_GSA_USAGE_CODE Enabled? Yes

PL/SQL File :

Purpose : .

Trigger Time: BEFORE Level: ROW Insert? Yes

Delete? No

New Alias NEW Update? No

Old Alias : OLD

Trigger Logic

```

DECLARE
--
--
tmp_rec_type_desc VARCHAR2(60);
--
--
tmp_use_code_desc VARCHAR2(60);
--
-- FETCH DESC FOR REC TYPE.
CURSOR GSA_REC_TYPE CUR(
search_rec_type ii_record_types.rec_type%type )
IS
SELECT description
FROM ii_record_types
WHERE rec_type = search_rec_type;
--
-- FETCH DESC FOR GSA USE CODE.
CURSOR GSA_USE_CODE CUR(
search_use_code ii_use_codes.use_code%type )
IS
SELECT description
FROM ii_use_codes
WHERE use_code = search_use_code;
BEGIN
BEGIN
IF :new.use_code IS NOT NULL and
:new.rec_type IS NOT NULL THEN
:new.usage_code := :new.rec_type || :new.use_code;
open gsa_use_code_cur (:new.use_code);
fetch gsa_use_code_cur into tmp_use_code_desc;
close gsa_use_code_cur;
open gsa_rec_type_cur (:new.rec_type);
fetch gsa_rec_type_cur into tmp_rec_type_desc;
close gsa_rec_type_cur;
:new.name :=
substr( tmp_use_code_desc||'-'|| tmp_rec_type_desc, 1,120);
END IF;
END;
END;
```

Oracle Designer Report : cktrigger

Container: Real Property Management

Version :

Table/View Name :LAND_UNITS (table)

Trigger Name :	II_TI_LU_TRG	Create?	Yes
PL/SQL Module			
Name :	II_TI_LU_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
New Alias	NEW	Delete?	No
Old Alias :	OLD	Update?	No

Trigger Logic

```

DECLARE
BEGIN
  BEGIN
    IF :new.cn IS NULL THEN
      :new.cn := db_instance.generate_global_id('LU_CN_SEQ');
    END IF;
    db_instance.insert_audit_columns
      (:new.created_by, :new.created_in_instance,
:new.created_date);
  END;
END;

```

Container: Real Property Management Version :

Table/View Name :II_BLDG_FLOORS_ROOMS (table)

Trigger Name : PUT_ID_AND_NAME Create? Yes

PL/SQL Module

Name : PUT_ID_AND_NAME Enabled? Yes

PL/SQL File :

Purpose : To capture bld id and bld name

Trigger Time: BEFORE Level: ROW Insert? Yes

Delete? No

New Alias

Update? No

Old Alias :

Trigger Logic

```
begin
  select id,name into :new.building_id,:new.building_name from
    ii_features where cn=:new.fea_cn;
end;
```

Container: Real Property Management Version :

Table/View Name :II_RANGE_PERMITS (table)

Trigger Name :	TI_RNGPMT	Create?	Yes
PL/SQL Module			
Name :	TI_RNGPMT	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
		Delete?	No
New Alias	NEW	Update?	No
Old Alias	OLD		

Trigger Logic

```

DECLARE
BEGIN
    -- Set the Created audit columns on row insert.
    db_instance.insert_audit_columns
    (:new.created_by,
     :new.created_in_instance,
     :new.created_date) ;
END;
```

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

Trigger Name :	II_LU_ASUD_TRG	Create? Yes
PL/SQL Module		
Name :	II_LU_ASUD_TRG	Enabled? Yes
PL/SQL File :		
Purpose :	After statement update/delete.	
Trigger Time:	AFTER	Level: STATEMENT
		Insert? No
		Delete? Yes
New Alias		Update? Yes
Old Alias :		

Trigger Logic

```

DECLARE
BEGIN
  -- Set the Created audit columns on row insert.
  db_instance.insert_audit_columns
    (:new.created_by,
     :new.created_in_instance,
     :new.created_date) ;
END;
Declare
  --
  vCN land_units.cn%TYPE := null;
  --
  -- Cursor to check for FK to Grazing Permit line item RMUs
  cursor cRMU is
    select 'x'
      from ii_range_permit_line_items
     where rmu_lu_cn = vCN; -- :old.cn;
  --
  -- Cursor to check for FK to Grazing Permit line item State IDs
  cursor cState is
    select 'x'
      from ii_range_permit_line_items
     where lu_state_cn = vCN; -- :old.cn;
  --
  -- Cursor to check for FK to Grazing Permit line item Proclaimed
  Unit/County IDs
  cursor cPUCnty is
    select 'x'
      from ii_range_permit_line_items
     where lu_proc_unit_cn = vCN; -- :old.cn;
  --
  vDummy varchar2(1);
  --
Begin
  --
  for vIndex in 1..ii_range_permits_audit.vNumEntries loop
    --
    vCN := ii_range_permits_audit.vPRAudit(vIndex).old_cn;
    --
    -- Check for FK to Grazing Permit line item RMUs.
    -- If found then write audit record. If not, then continue
    open cRMU;

```

Oracle Designer Report : ckttrigger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

        fetch cRMU into vDummy;
        if cRMU%found then
            close cRMU;
            ii_range_permits_audit.Write_Grz_Permit_LI_LU_Audits (
ii_range_permits_audit.vPRAudit(vIndex).old_cn,

ii_range_permits_audit.vPRAudit(vIndex).old_id,

ii_range_permits_audit.vPRAudit(vIndex).new_id,

'RPLI_RM_UK' );
        else
            close cRMU;
        end if;
        --
        -- Check for FK to Grazing Permit line item RMUs.
        -- If found then write audit record. If not, then continue
        open cState;
        fetch cState into vDummy;
        if cState%found then
            close cState;
            ii_range_permits_audit.Write_Grz_Permit_LI_LU_Audits (
ii_range_permits_audit.vPRAudit(vIndex).old_cn,

ii_range_permits_audit.vPRAudit(vIndex).old_id,

ii_range_permits_audit.vPRAudit(vIndex).new_id,

'RPLI_LU_STATE_FK' );
        else
            close cState;
        end if;
        --
        -- Check for FK to Grazing Permit line item Proclaimed
        Unit/County IDs.
        -- If found then write audit record. If not, then continue
        open cPUCnty;
        fetch cPUCnty into vDummy;
        if cPUCnty%found then
            close cPUCnty;
            ii_range_permits_audit.Write_Grz_Permit_LI_LU_Audits (
ii_range_permits_audit.vPRAudit(vIndex).old_cn,

ii_range_permits_audit.vPRAudit(vIndex).old_id,

ii_range_permits_audit.vPRAudit(vIndex).new_id,

'RPLI_LU_PROC_UNIT_FK' );
        else
            close cPUCnty;
        end if;
        --
    end loop;
    --
    -- Re-set counter so that next update statement in this session
    works correctly.
    ii_range_permits_audit.vNumEntries := 0;
    --
End;
```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :II_CONTACTS (table)

Trigger Name : II_CONT_ASUD_TRG Create? Yes
 PL/SQL Module
 Name : II_CONT_ASUD_TRG Enabled? Yes
 PL/SQL File :
 Purpose : After update/delete statement-level trigger.
 Trigger Time: AFTER Level: STATEMENT Insert? No
 Delete? Yes
 New Alias Update? Yes
 Old Alias :

Trigger Logic

```

Declare
--
  vCN land_units.cn%TYPE := null;
--
-- Cursor to check for FK to Grazing Permit line item RMUs
  cursor cRMU is
    select 'x'
      from ii_range_permit_line_items
     where rmu_lu_cn = vCN; --:old.cn;
--
-- Cursor to check for FK to Grazing Permit line item State IDs
  cursor cState is
    select 'x'
      from ii_range_permit_line_items
     where lu_state_cn = vCN; -- :old.cn;
--
-- Cursor to check for FK to Grazing Permit line item Proclaimed
Unit/County IDs
  cursor cPUCnty is
    select 'x'
      from ii_range_permit_line_items
     where lu_proc_unit_cn = vCN; -- :old.cn;
--
  vDummy varchar2(1);
--
Begin
--
  for vIndex in 1..ii_range_permits_audit.vNumEntries loop
    --
    vCN := ii_range_permits_audit.vPRAudit(vIndex).old_cn;
    --
    -- Check for FK to Grazing Permit line item RMUs.
    -- If found then write audit record. If not, then continue
    open cRMU;
    fetch cRMU into vDummy;
    if cRMU%found then
      close cRMU;
      ii_range_permits_audit.Write_Grz_Permit_LI_LU_Audits (
ii_range_permits_audit.vPRAudit(vIndex).old_cn,
ii_range_permits_audit.vPRAudit(vIndex).old_id,

```

Oracle Designer Report : cktrigger

Container: Real Property Management Version :

Table/View Name :II_CONTACTS (table)

```

ii_range_permits_audit.vPRAudit(vIndex).new_id,
'RPLI_RMJ_FK' );
    else
        close cRMJ;
    end if;
    --
    -- Check for FK to Grazing Permit line item RMJs.
    -- If found then write audit record. If not, then continue
    open cState;
    fetch cState into vDummy;
    if cState%found then
        close cState;
        ii_range_permits_audit.Write_Grz_Permit_LI_LU_Audits (
ii_range_permits_audit.vPRAudit(vIndex).old_cn,
ii_range_permits_audit.vPRAudit(vIndex).old_id,
ii_range_permits_audit.vPRAudit(vIndex).new_id,
'RPLI_LU_STATE_FK' );
    else
        close cState;
    end if;
    --
    -- Check for FK to Grazing Permit line item Proclaimed
Unit/County IDs.
    -- If found then write audit record. If not, then continue
    open cPUCnty;
    fetch cPUCnty into vDummy;
    if cPUCnty%found then
        close cPUCnty;
        ii_range_permits_audit.Write_Grz_Permit_LI_LU_Audits (
ii_range_permits_audit.vPRAudit(vIndex).old_cn,
ii_range_permits_audit.vPRAudit(vIndex).old_id,
ii_range_permits_audit.vPRAudit(vIndex).new_id,
'RPLI_LU_PROC_UNIT_FK' );
    else
        close cPUCnty;
    end if;
    --
    end loop;
    --
    -- Re-set counter so that next update statement in this session
works correctly.
    ii_range_permits_audit.vNumEntries := 0;
    --
End;
```

Container: Real Property Management Version :

Table/View Name :II_CONTACTS (table)

```

        select 'x'
          from ii_range_permits
         where permittee_cn = vCN; -- :old.cn;
--
-- Cursor to check for FK to Grazing Permit managing organization
cursor cMOrg is
  select 'x'
    from accomplishment_instruments
   where managing_cont_cn = vCN and      -- :old.cn and
         obj_name         = 'RANGE_PERMIT';
--
vDummy varchar2(1);
--
Begin
--
  for vIndex in 1..ii_range_permits_audit.vNumEntries loop
    --
    vCN := ii_range_permits_audit.vPRAudit(vIndex).old_cn;
    --
    -- Check for FK to Grazing Permit permittees.
    -- If found then write audit record. If not, then continue
    open cPmtee;
    fetch cPmtee into vDummy;
    if cPmtee%found then
      close cPmtee;
      ii_range_permits_audit.Write_Grz_Pmt_Contact_Audits (
ii_range_permits_audit.vPRAudit(vIndex).old_cn,

ii_range_permits_audit.vPRAudit(vIndex).old_id,

ii_range_permits_audit.vPRAudit(vIndex).new_id,

'RNGPMT_CONT_CN' );
    else
      close cPmtee;
    end if;
    --
    -- Check for FK to Grazing Permit managing orgs.
    -- If found then write audit record. If not, then continue
    open cMOrg;
    fetch cMOrg into vDummy;
    if cMOrg%found then
      close cMOrg;
      ii_range_permits_audit.Write_Grz_Pmt_Contact_Audits (
ii_range_permits_audit.vPRAudit(vIndex).old_cn,

ii_range_permits_audit.vPRAudit(vIndex).old_id,

ii_range_permits_audit.vPRAudit(vIndex).new_id,

'ACCINST_CONT_FK' );
    else
      close cMOrg;
    end if;
    --
  end loop;
--
-- Re-set counter so that next update statement in this session
works correctly.

```

Oracle Designer Report : cktrgger

Container: Real Property Management **Version :**

Table/View Name :II_CONTACTS (table)

```
    ii_range_permits_audit.vNumEntries := 0;  
    --  
End;
```

Container: Real Property Management Version :

Table/View Name :II_RANGE_PERMITS (table)

Trigger Name :	TU_RNGPMT	Create?	Yes
PL/SQL Module			
Name :	TU_RNGPMT	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	No
New Alias	NEW	Delete?	No
Old Alias :	OLD	Update?	Yes

Trigger Logic

```

DECLARE
BEGIN
  -- Set the Modified audit columns when updating a row.
  db_instance.update_audit_columns
    (:new.modified_by,
     :new.modified_in_instance,
     :new.modified_date ) ;
END;
```

Container: Real Property Management Version :

Table/View Name :II_PERSONS (table)

Trigger Name :	II_TIU_PERSONS	Create?	Yes
PL/SQL Module			
Name :	II_TIU_PERSONS	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
		Delete?	No
New Alias	NEW	Update?	Yes
Old Alias	: OLD		

Trigger Logic

```

DECLARE
BEGIN
  -- Check privileges to insert/update SSN.
  if not ii_chk_role_assign('II_SECURITY_OFFICER') then
    if INSERTING then
      if :new.social_security_number is not null then
        raise_application_error(-20301, 'Insufficient privileges
to '||
                                'insert SSN(s) for any Person(s)');
      end if;
    else -- updating
      if nvl(:old.social_security_number,'@')
        != nvl(:new.social_security_number,'@') then
        raise_application_error(-20302, 'Insufficient privileges
to '||
                                'update SSN(s) for any Person(s)');
      end if;
    end if;
  end if;
END;
```

Container: Real Property Management

Version :

Table/View Name :II_LEASES (table)

Trigger Name :	LEASE_IICODE_TRG	Create?	Yes
PL/SQL Module			
Name :	LEASE_IICODE_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
		Delete?	No
New Alias	NEW	Update?	Yes
Old Alias	: OLD		

Trigger Logic

```

DECLARE
BEGIN
  -- Check privileges to insert/update SSN.
  if not ii_chk_role_assign('II_SECURITY_OFFICER') then
    if INSERTING then
      if :new.social_security_number is not null then
        raise_application_error(-20301, 'Insufficient privileges
to '||
                                'insert SSN(s) for any Person(s)');
      end if;
    else -- updating
      if nvl(:old.social_security_number,'@')
        != nvl(:new.social_security_number,'@') then
        raise_application_error(-20302, 'Insufficient privileges
to '||
                                'update SSN(s) for any Person(s)');
      end if;
    end if;
  end if;
END;
DECLARE
BEGIN
  --
  IF :new.STATUS_CODE is not null and
    :new.STATUS_CODE <> nvl(:old.STATUS_CODE,'@') and
    not Chk_iicode( 'LEASE_STATUS', :new.STATUS_CODE )
  THEN
    Raise_Application_Error( -20302,
      'Invalid value "'|| :new.STATUS_CODE ||
      ' for ii_Leases.STATUS_CODE.' );
  END IF;
  --
  IF :new.SERVICE_CODE is not null and
    :new.SERVICE_CODE <> nvl(:old.SERVICE_CODE,'@') and
    not Chk_iicode( 'LEASE_SERVICE', :new.SERVICE_CODE )
  THEN
    Raise_Application_Error( -20303,
      'Invalid value "'|| :new.SERVICE_CODE ||
      ' for ii_Leases.SERVICE_CODE.' );
  END IF;
  --

```

Oracle Designer Report : ckttrigger

Container: Real Property Management Version :

Table/View Name :II_LEASES (table)

```

--
IF :new.ESCALATOR is not null and
:new.ESCALATOR <> nvl(:old.ESCALATOR,'@') and
not II_Chk_615RefCode( 'YESNO', :new.ESCALATOR )
THEN
  Raise_Application_Error( -20305,
    'Invalid value "'|| :new.ESCALATOR ||'" '||
    'for II_Leases.ESCALATOR.' );
END IF;
--
IF nvl(:new.NUMBER_OF_OPTIONS,0) not between 0 and 999
THEN
  Raise_Application_Error( '20306',
    'II_Leases.NUMBER_OF_OPTIONS must be between 0 and 999.' );
END IF;
--
IF nvl(:new.TERMINATION_NOTICE_DAYS,0) not between 0 and 999
THEN
  Raise_Application_Error( '20305',
    'II_Leases.TERMINATION_NOTICE_DAYS must be between 0 and
999.' );
END IF;
--
IF nvl(:new.OPTION_PERIOD,0) not between 0 and 999
THEN
  Raise_Application_Error( '20305',
    'II_Leases.OPTION_PERIOD must be between 0 and 999.' );
END IF;
--
IF nvl(:new.OPTION_NOTICE_DAYS,0) not between 0 and 999
THEN
  Raise_Application_Error( '20305',
    'II_Leases.OPTION_NOTICE_DAYS must be between 0 and 999.' );
END IF;
--
IF nvl(:new.MONTHLY_RENTAL_AMOUNT,0) < 0 or
nvl(:new.ANNUAL_RENTAL_AMOUNT,0) < 0 or
nvl(:new.BASE_TAX,0) < 0 or
nvl(:new.BASE_OPERATING_COST,0) < 0
THEN
  Raise_Application_Error( '20306',
    'Values for BASE and AMOUNT columns in II_Leases must be >
0.' );
END IF;
--
END;

```

Container: Real Property Management

Version :

Table/View Name :II_ADDRESSES (table)

Trigger Name :	ADD_IICODE_TRG	Create?	Yes
PL/SQL Module			
Name :	ADD_IICODE_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
New Alias	NEW	Delete?	No
Old Alias :	OLD	Update?	Yes

Trigger Logic

```

DECLARE
BEGIN
    if (:new.ADDRESS_TYPE is not null
        and not chk_iicode('ADDRESS_TYPE',:new.ADDRESS_TYPE))
    then
        raise_application_error(-20300, 'Invalid value; II_CODES name not
found.');
```


Container: Real Property Management Version :

Table/View Name : II_LU_LU_LINKS (table)

Trigger Name :	II_TIU_LULULNK_TRG	Create?	Yes
PL/SQL Module			
Name :	II_TIU_LULULNK_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
		Delete?	No
New Alias	NEW	Update?	Yes
Old Alias	OLD		

Trigger Logic

```

DECLARE
BEGIN
    if (:new.ADDRESS_TYPE is not null
        and not chk_ilcode('ADDRESS_TYPE',:new.ADDRESS_TYPE))
    then
        raise_application_error(-20300, 'Invalid value; II_CODES name not
found.');
```

```

    end if;
END;
DECLARE
BEGIN
    if (:new.ADDRESS_TYPE is not null
        and not chk_ilcode('ADDRESS_TYPE',:new.ADDRESS_TYPE))
    then
        raise_application_error(-20300, 'Invalid value; II_CODES name not
found.');
```

```

    end if;
END;
DECLARE
BEGIN
    if (:new.ADDRESS_TYPE is not null
        and not chk_ilcode('ADDRESS_TYPE',:new.ADDRESS_TYPE))
    then
        raise_application_error(-20300, 'Invalid value; II_CODES name not
found.');
```

```

    end if;
END;
DECLARE
BEGIN
    DECLARE
        vFromLUCN ii_lu_lu_links.from_lu_cn%TYPE := null;
        vToLUCN ii_lu_lu_links.to_lu_cn%TYPE := null;
        cursor cFrom is
            select obj_name
            from land_units
            where cn = vFromLUCN;
        cursor cTo is
            select obj_name
            from land_units
            where cn = vToLUCN;
        vFromObjName land_units.obj_name%TYPE := null;
```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :II_LU_LU_LINKS (table)

```

vToObjName land_units.obj_name%TYPE := null;
BEGIN
-- Check for valid separation distance uom.
IF (:new.SEPARATION_DISTANCE_UOM is not null
and not
chk_iicode('DISTANCE_UOM',:new.SEPARATION_DISTANCE_UOM))
THEN
raise_application_error(-20301, 'Invalid value "'||
:new.SEPARATION_DISTANCE_UOM||
'" for Separation Distance Unit - Please re-enter');
END IF;
-- Check for area uom.
IF (:new.AREA_UOM is not null
and not chk_iicode('AREA_UOM',:new.AREA_UOM))
THEN
raise_application_error(-20310, 'Invalid value "'||
:new.AREA_UOM||
'" for Area Unit - Please re-enter');
END IF;
-- Check role assignment for Wilderness-Geolocator updates.
IF not ii_chk_role_assign('INFRA_SVC_MGR') THEN
IF INSERTING THEN
vFromLUCN := :new.from_lu_cn;
vToLUCN := :new.to_lu_cn;
open cFrom;
fetch cFrom into vFromObjName;
close cFrom;
open cTo;
fetch cTo into vToObjName;
close cTo;
IF vFromObjName = 'WILDERNESS' THEN
IF vToObjName in
('STATE','COUNTY','CONGRESSIONAL_DISTRICT',
'PROCLAIMED_UNIT') then
raise_application_error(-20302,'Insufficient
privileges to '||
'insert a Wilderness-'||
initcap(replace(vToObjName,'_',' '))||'
link');
END IF;
ELSIF vFromObjName in
('STATE','COUNTY','CONGRESSIONAL_DISTRICT',
'PROCLAIMED_UNIT') then
IF vToObjName = 'WILDERNESS' then
raise_application_error(-20302,'Insufficient
privileges to '||
'insert a Wilderness-'||
initcap(replace(vFromObjName,'_',' '))||'
link');
END IF;
ELSE -- Updating
-- Updating non-CN column(s).
vFromLUCN := :old.from_lu_cn;
vToLUCN := :old.to_lu_cn;
open cFrom;
fetch cFrom into vFromObjName;
close cFrom;
open cTo;

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :II_LU_LU_LINKS (table)

```

        fetch cTo into vToObjName;
        close cTo;
        IF vFromObjName = 'WILDERNESS' THEN
            IF vToObjName in
                ('STATE','COUNTY','CONGRESSIONAL_DISTRICT',
                 'PROCLAIMED_UNIT') then
                raise_application_error(-20302,'Insufficient
privileges to '||
                'update a Wilderness-'||
                initcap(replace(vToObjName,'_',' '))||' link');
            END IF;
        ELSIF vFromObjName in
            ('STATE','COUNTY','CONGRESSIONAL_DISTRICT',
             'PROCLAIMED_UNIT') then
            IF vToObjName = 'WILDERNESS' then
                raise_application_error(-20302,'Insufficient
privileges to '||
                'update a Wilderness-'||
                initcap(replace(vFromObjName,'_',' '))||'
link');
            END IF;
        END IF;
        -- Changed "from" CN (+ other column(s)).
        IF (:new.from_lu_cn != :old.from_lu_cn) THEN
            vFromLUCN := :new.from_lu_cn;
            vToLUCN := :old.to_lu_cn;
            open cFrom;
            fetch cFrom into vFromObjName;
            close cFrom;
            open cTo;
            fetch cTo into vToObjName;
            close cTo;
            IF vFromObjName = 'WILDERNESS' THEN
                IF vToObjName in
                    ('STATE','COUNTY','CONGRESSIONAL_DISTRICT',
                     'PROCLAIMED_UNIT') then
                    raise_application_error(-20302,
                    'Insufficient privileges to '||
                    'create a Wilderness-'||
                    initcap(replace(vToObjName,'_',' '))||' link');
                END IF;
            ELSIF vFromObjName in
                ('STATE','COUNTY','CONGRESSIONAL_DISTRICT',
                 'PROCLAIMED_UNIT') then
                IF vToObjName = 'WILDERNESS' then
                    raise_application_error(-20302,
                    'Insufficient privileges to '||
                    'create a Wilderness-'||
                    initcap(replace(vFromObjName,'_',' '))||'
link');
                END IF;
            END IF;
        -- Changed "to" CN (+ other column(s)).
        IF (:new.to_lu_cn != :old.to_lu_cn) THEN
            vFromLUCN := :old.from_lu_cn;
            vToLUCN := :new.to_lu_cn;
            open cFrom;
            fetch cFrom into vFromObjName;

```

Oracle Designer Report : cktggr

Container: Real Property Management Version :

Table/View Name :II_LU_LU_LINKS (table)

```

close cFrom;
open cTo;
fetch cTo into vToObjName;
close cTo;
IF vFromObjName = 'WILDERNESS' THEN
  IF vToObjName in
    ('STATE','COUNTY','CONGRESSIONAL_DISTRICT',
     'PROCLAIMED_UNIT') then
    raise application_error(-20302,
      'Insufficient privileges to '||
      'create a Wilderness-'||
      initcap(replace(vToObjName,'_',' '))||' link');
  END IF;
ELSIF vFromObjName in
  ('STATE','COUNTY','CONGRESSIONAL_DISTRICT',
   'PROCLAIMED_UNIT') then
  IF vToObjName = 'WILDERNESS' then
    raise application_error(-20302,
      'Insufficient privileges to '||
      'create a Wilderness-'||
      initcap(replace(vFromObjName,'_',' '))||'
link');
  END IF;
END IF;
END IF;
END IF;
END IF;
END;
END;

```

Container: Real Property Management

Version :

Table/View Name :II_BLDG_DOORS (table)

Trigger Name :	BLDDR_IICODE_TRG	Create?	Yes
PL/SQL Module			
Name :	BLDDR_IICODE_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
New Alias	NEW	Delete?	No
Old Alias	OLD	Update?	Yes

Trigger Logic

```

DECLARE
BEGIN
    if (:new.DOOR_TYPE is not null
        and not chk_iicode('DOOR_TYPE',:new.DOOR_TYPE))
    or (:new.DOOR_MATERIAL is not null
        and not chk_iicode('DOOR_MATERIAL',:new.DOOR_MATERIAL))
    or (:new.SECONDARY_MATERIAL is not null
        and not chk_iicode('DOOR_MATERIAL',:new.SECONDARY_MATERIAL))
    or (:new.TERTIARY_MATERIAL is not null
        and not chk_iicode('DOOR_MATERIAL',:new.TERTIARY_MATERIAL))
    or (:new.CONDITION_RATING is not null
        and not
chk_iicode('FEAT_CONDITION_RATING',:new.CONDITION_RATING))
    or (:new.APPRAISAL_RATING is not null
        and not
chk_iicode('FEAT APPRAISAL_RATING',:new.APPRAISAL_RATING))
    or (:new.ACCESSIBILITY_STATUS is not null
        and not
chk_iicode('ACCESSIBILITY_STATUS',:new.ACCESSIBILITY_STATUS))
    then
        raise_application_error(-20300, 'Invalid value; II_CODES name not
found.');
```

```

    end if;
END;
```

Container: Real Property Management Version :

Table/View Name :II_DOC_ACCINST_LINKS (table)

Trigger Name :	II_DOCACCI_TIU_TRG	Create?	Yes
PL/SQL Module			
Name :	II_DOCACCI_TIU_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
New Alias	NEW	Delete?	No
Old Alias	OLD	Update?	Yes

Trigger Logic

```

DECLARE
BEGIN
  BEGIN
    IF INSERTING THEN
      db_instance.insert_audit_columns
        (:new.created_by, :new.created_in_instance,
:new.created_date);
    ELSE
      db_instance.update_audit_columns
        (:new.modified_by, :new.modified_in_instance,
:new.modified_date);
    END IF;
  END;
END;
```

Container: Real Property Management Version :

Table/View Name :II_LEASE_NOTIFICATIONS (table)

Trigger Name : II_RBIU_LEASE_NOTIFICATIONS Create? Yes

PL/SQL Module

Name : II_RBIU_LEASE_NOTIFICATIONS Enabled? Yes

PL/SQL File :

Purpose : .

Trigger Time: BEFORE Level: ROW Insert? Yes

Delete? No

New Alias NEW Update? Yes

Old Alias : OLD

Trigger Logic

```

DECLARE
BEGIN
  IF :new.NOTIFICATION EMAILED <> nvl(:old.NOTIFICATION EMAILED, '@')
and
  NOT II_Chk_615RefCode( 'YESNO', :new.NOTIFICATION EMAILED )
  THEN
    Raise_Application_Error( -20201,
      'Invalid value "' || :new.NOTIFICATION EMAILED || '" ' ||
      'for II_Lease_Notifications.NOTIFICATION EMAILED.' );
  END IF;
  --
  IF :new.EMAIL_ADDRESS not like '%@%'
  THEN
    Raise_Application_Error( -20202,
      'Email addresses must be formatted as name@organization.com'
    );
  END IF;
  --
END;
```

Container: Real Property Management Version :

Table/View Name :II_BLDG_EQUIPMENT (table)

Trigger Name :	BLDEQ_IICODE_TRG	Create?	Yes
PL/SQL Module			
Name :	BLDEQ_IICODE_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
		Delete?	No
New Alias	NEW	Update?	Yes
Old Alias	: OLD		

Trigger Logic

```

DECLARE
BEGIN
    if (:new.CONDITION_RATING is not null
        and not
        chk_iicode('FEAT_CONDITION_RATING',:new.CONDITION_RATING))
        or (:new.APPRAISAL_RATING is not null
            and not
            chk_iicode('FEAT APPRAISAL_RATING',:new.APPRAISAL_RATING))
        or (:new.FUEL_ENERGY is not null
            and not chk_iicode('FUEL_ENERGY',:new.FUEL_ENERGY))
        then
            raise_application_error(-20300, 'Invalid value; II_CODES name not
found.');
```


Container: Real Property Management Version :

Table/View Name :II_LINK_TYPES (table)

Trigger Name :	TIU_IILTYP	Create?	Yes
PL/SQL Module			
Name :	TIU_IILTYP	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
New Alias	NEW	Delete?	No
Old Alias :	OLD	Update?	Yes

Trigger Logic

```

DECLARE
BEGIN
  IF inserting
  THEN
    db_instance.insert_audit_columns(:new.created_by,
    :new.created_in_instance,
    :new.created_date);
  ELSE db_instance.update_audit_columns(:new.modified_by,
    :new.modified_in_instance,
    :new.modified_date);
  END IF;
END;
```

Container: Real Property Management

Version :

Table/View Name :II_BUILDINGS (table)

Trigger Name : BLD_IICODE_TRG Create? Yes

PL/SQL Module

Name : BLD_IICODE_TRG Enabled? Yes

PL/SQL File :

Purpose : .

Trigger Time: BEFORE Level: ROW Insert? Yes

Delete? No

New Alias NEW Update? Yes

Old Alias : OLD

Trigger Logic

```
DECLARE
BEGIN
  if (:new.SUBCATEGORY is not null
      and not chk_iicode('SUBCATEGORY', :new.SUBCATEGORY))
  then
    raise_application_error(-20300, 'Invalid value; II_CODES name
not found.');
```

```
  end if;
```

```
END;
```

Container: Real Property Management Version :

Table/View Name :II_LU_CONT_LINKS (table)

Trigger Name : II_TIU_LUCONT_TRG Create? Yes

PL/SQL Module

Name : II_TIU_LUCONT_TRG Enabled? Yes

PL/SQL File :

Purpose : .

Trigger Time: BEFORE Level: ROW Insert? Yes

Delete? No

New Alias NEW Update? Yes

Old Alias : OLD

Trigger Logic

```

DECLARE
BEGIN
  if (:new.SUBCATEGORY is not null
      and not chk_iicode('SUBCATEGORY', :new.SUBCATEGORY))
  then
    raise_application_error(-20300, 'Invalid value; II_CODES name
not found.');
```

```

  end if;
END;
DECLARE
BEGIN
  DECLARE
    vLUCN ii_lu_cont_links.lu_cn%TYPE := null;
    cursor cWild is
      select obj_name
      from land_units
      where cn = vLUCN;
    vLUObjName land_units.obj_name%TYPE := null;
  BEGIN
    -- Check for valid area uom.
    IF (:new.AREA_UOM is not null
        and not chk_iicode('AREA_UOM', :new.AREA_UOM))
    THEN
      raise_application_error(-20301, 'Invalid value "' ||
        :new.AREA_UOM ||
        '" for Area Unit - Please re-enter');
    END IF;
    -- Check role assignment for Wilderness-Admin. Unit updates.
    IF not ii_chk_role_assign('INFRA_SVC_MGR') THEN
      --
      IF INSERTING THEN
        --
        vLUCN := :new.lu_cn;
        open cWild;
        fetch cWild into vLUObjName;
        close cWild;
        IF vLUObjName = 'WILDERNESS' THEN
          IF :new.link_type_name = 'ADMINISTRATIVE_UNIT' THEN
            raise_application_error(-20302, 'Insufficient
privileges to ' ||
              'insert a Wilderness-Administrative Unit link');
          END IF;

```

Oracle Designer Report : ckttrigger

Container: Real Property Management Version :

Table/View Name :II_LU_CONT_LINKS (table)

```

        END IF;
    --
    ELSE -- Updating
    --
        -- Updating non-CN column(s).
        vLUCN := :old.lu_cn;
        open cWild;
        fetch cWild into vLUObjName;
        close cWild;
        IF vLUObjName = 'WILDERNESS' THEN
            IF :old.link_type_name = 'ADMINISTRATIVE_UNIT' THEN
                raise_application_error(-20303, 'Insufficient
privileges to '||
                'update a Wilderness-Administrative Unit link');
            END IF;
        END IF;
        -- Changed Land Unit CN (+ other column(s)).
        IF (:new.lu_cn != :old.lu_cn) THEN
            vLUCN := :new.lu_cn;
            open cWild;
            fetch cWild into vLUObjName;
            close cWild;
            IF vLUObjName = 'WILDERNESS' THEN
                IF :new.link_type_name = 'ADMINISTRATIVE_UNIT' THEN
                    raise_application_error(-20304,
                        'Insufficient privileges to '||
                        'create a Wilderness-Administrative Unit
link');
                END IF;
            END IF;
        END IF;
        -- Changed Link Type Name to 'ADMINISTRATIVE_UNIT'
        IF (:new.link_type_name = 'ADMINISTRATIVE_UNIT') AND
        (:new.link_type_name != :old.link_type_name) THEN
            vLUCN := :old.lu_cn;
            open cWild;
            fetch cWild into vLUObjName;
            close cWild;
            IF vLUObjName = 'WILDERNESS' THEN
                raise_application_error(-20304,
                    'Insufficient privileges to '||
                    'create a Wilderness-Administrative Unit
link');
            END IF;
        END IF;
    END IF;
END;
END;

```

Container: Real Property Management Version :

Table/View Name :II_BLDG_FLOORS (table)

Trigger Name :	BLDFL_IICODE_TRG	Create?	Yes
PL/SQL Module			
Name :	BLDFL_IICODE_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
		Delete?	No
New Alias	NEW	Update?	Yes
Old Alias	: OLD		

Trigger Logic

```

DECLARE
BEGIN
    if (:new.ACCESSIBILITY_STATUS is not null
        and not
chk_iicode('ACCESSIBILITY_STATUS',:new.ACCESSIBILITY_STATUS))
        or (:new.TYPE_OF_CONSTRUCTION is not null
            and not
chk_iicode('TYPE_OF_CONSTRUCTION',:new.TYPE_OF_CONSTRUCTION))
        then
            raise_application_error(-20300, 'Invalid value; II_CODES name not
found.');
```

Container: Real Property Management Version :

Table/View Name :II_PHONE_NUMBERS (table)

Trigger Name :	PN_IICODE_TRG	Create?	Yes
PL/SQL Module			
Name :	PN_IICODE_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
New Alias	NEW	Delete?	No
Old Alias :	OLD	Update?	Yes

Trigger Logic

```

DECLARE
BEGIN
  if (:new.PHONE_NUMBER_TYPE is not null
    and not chk_iicode('PHONE_NUMBER_TYPE',:new.PHONE_NUMBER_TYPE))
  then
    raise_application_error(-20300, 'Invalid value; II_CODES name not
found.');
```

```

  end if;
END;
```

Container: Real Property Management Version :

Table/View Name :II_LEASE_LINE_ITEMS (table)

Trigger Name : II_LEASE_LINE_ITEM_TRIG Create? Yes

PL/SQL Module

Name : II_LEASE_LINE_ITEM_TRIG Enabled? Yes

PL/SQL File :

Purpose : .

Trigger Time: BEFORE Level: ROW Insert? Yes

Delete? No

New Alias NEW Update? Yes

Old Alias : OLD

Trigger Logic

```
DECLARE
BEGIN
  IF INSERTING THEN
    db_instance.insert_audit_columns
      (:new.created_by, :new.created_in_instance,
:new.created_date);
  ELSE
    db_instance.update_audit_columns
      (:new.modified_by, :new.modified_in_instance,
:new.modified_date);
  END IF;
END;
```

Container: Real Property Management Version :

Table/View Name :II_LEASE_LINE_ITEMS (table)

Trigger Name :	LLI_IICODE_TRG	Create?	Yes
PL/SQL Module			
Name :	LLI_IICODE_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
		Delete?	No
New Alias	NEW	Update?	Yes
Old Alias	OLD		

Trigger Logic

```

DECLARE
BEGIN
  IF :new.UOM_CODE is not null and
    :new.UOM_CODE <> nvl(:old.UOM_CODE,'@') and
    not Chk_iicode( 'LEASE_UOM', :new.UOM_CODE )
  THEN
    Raise_Application_Error( -20300,
      'Invalid value "'||:new.UOM_CODE||'" '||
      'for ii_Lease_Line_Items.UOM_CODE.' );
  END IF;
  --
  /*
  IF :new.LAND_UNIT_2250A is not null and
    :new.LAND_UNIT_2250A <> nvl(:old.LAND_UNIT_2250A,'@') and
    not II_Chk_615RefCode( 'YESNO', :new.LAND_UNIT_2250A )
  THEN
    Raise_Application_Error( -20305,
      'Invalid value "'|| :new.LAND_UNIT_2250A ||'" '||
      'for II_Lease_Line_Items.LAND_UNIT_2250A.' );
  END IF;
  */
  --
END;
```


Container: Real Property Management Version :

Table/View Name :II_CONT_CONT_LINKS (table)

Trigger Name :	II_TIU_CONTLNK_TRG	Create?	Yes
PL/SQL Module			
Name :	II_TIU_CONTLNK_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
		Delete?	No
New Alias	NEW	Update?	Yes
Old Alias	: OLD		

Trigger Logic

```

DECLARE
BEGIN
    BEGIN
        IF INSERTING THEN
            db_instance.insert_audit_columns
                (:new.created_by, :new.created_in_instance,
:new.created_date);
        ELSE
            db_instance.update_audit_columns
                (:new.modified_by, :new.modified_in_instance,
:new.modified_date);
        END IF;
    END;
END;

```

Container: Real Property Management

Version :

Table/View Name :II_GRANTS (table)

Trigger Name :	T_RBIU_GRANTS	Create?	Yes
PL/SQL Module			
Name :	T_RBIU_GRANTS	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
		Delete?	No
New Alias	NEW	Update?	Yes
Old Alias	OLD		

Trigger Logic

```

DECLARE
BEGIN
  BEGIN
    IF inserting
    THEN
      -- Create the CN if not supplied on row insert.
      IF :new.CN is null
      THEN
        :new.CN := Db_Instance.GENERATE_GLOBAL_ID( 'II_GA_SEQ' );
      END IF;
      --
      -- Set the CREATED... audit columns on row insert if not
      supplied.
      IF :new.CREATED_BY is null or
        :new.CREATED_DATE is null or
        :new.CREATED_IN_INSTANCE is null
      THEN
        Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
                                          , :new.CREATED_IN_INSTANCE
                                          , :new.CREATED_DATE );
      END IF;
      ---
    ELSIF updating
    THEN
      -- Set the MODIFIED... audit columns on row update.
      Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY
                                       , :new.MODIFIED_IN_INSTANCE
                                       , :new.MODIFIED_DATE );
      ---
    END IF;
  END;
END;

```

Container: Real Property Management Version :

Table/View Name :II_FEA_CONDITIONS (table)

Trigger Name :	II_RBIU_FEA_COND	Create?	Yes
PL/SQL Module			
Name :	II_RBIU_FEA_COND1	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
		Delete?	No
New Alias		Update?	Yes
Old Alias :			

Trigger Logic

```

DECLARE
BEGIN
  -- Set the CREATED... audit columns on row insert if not supplied.
  IF :new.CREATED_BY is null or
     :new.CREATED_DATE is null or
     :new.CREATED_IN_INSTANCE is null
  THEN
    Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
                                     , :new.CREATED_IN_INSTANCE
                                     , :new.CREATED_DATE );
  END IF;
  ---
  ELSIF Updating
  THEN
    -- Set the MODIFIED... audit columns on row update.
    Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY
                                     , :new.MODIFIED_IN_INSTANCE
                                     , :new.MODIFIED_DATE );
    ---
  END IF;
END;
```

Container: Real Property Management Version :

Table/View Name :II_RANGE_PERMITS_V (view)

Trigger Name : II_TIUD_RNGPMTV_IO_TRG Create? Yes
 PL/SQL Module
 Name : II_TIUD_RNGPMTV_IO_TRG Enabled? Yes
 PL/SQL File :
 Purpose : .
 Trigger Time: INSTEAD OF Level: STATEMENT Insert? Yes
 Delete? Yes
 New Alias Update? Yes
 Old Alias :

Trigger Logic

```
DECLARE
BEGIN
  -- Set the CREATED... audit columns on row insert if not supplied.
  IF :new.CREATED_BY is null or
     :new.CREATED_DATE is null or
     :new.CREATED_IN_INSTANCE is null
  THEN
    Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
                                     , :new.CREATED_IN_INSTANCE
                                     , :new.CREATED_DATE );
  END IF;
  ---
  ELSIF Updating
  THEN
    -- Set the MODIFIED... audit columns on row update.
    Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY
                                     , :new.MODIFIED_IN_INSTANCE
                                     , :new.MODIFIED_DATE );
  ---
  END IF;
END;
DECLARE
BEGIN
  -- Set the CREATED... audit columns on row insert if not supplied.
  IF :new.CREATED_BY is null or
     :new.CREATED_DATE is null or
     :new.CREATED_IN_INSTANCE is null
  THEN
    Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
                                     , :new.CREATED_IN_INSTANCE
                                     , :new.CREATED_DATE );
  END IF;
  ---
  ELSIF Updating
  THEN
    -- Set the MODIFIED... audit columns on row update.
    Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY
                                     , :new.MODIFIED_IN_INSTANCE
                                     , :new.MODIFIED_DATE );
  ---
  END IF;
END;
```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :II_RANGE_PERMITS_V (view)

```

DECLARE
BEGIN
  -- Set the CREATED... audit columns on row insert if not supplied.
  IF :new.CREATED_BY is null or
     :new.CREATED_DATE is null or
     :new.CREATED_IN_INSTANCE is null
  THEN
    Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
                                     , :new.CREATED_IN_INSTANCE
                                     , :new.CREATED_DATE );
  END IF;
  ---
  ELSIF Updating
  THEN
    -- Set the MODIFIED... audit columns on row update.
    Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY
                                     , :new.MODIFIED_IN_INSTANCE
                                     , :new.MODIFIED_DATE );
    ---
  END IF;
END;
DECLARE
BEGIN
  -- Set the CREATED... audit columns on row insert if not supplied.
  IF :new.CREATED_BY is null or
     :new.CREATED_DATE is null or
     :new.CREATED_IN_INSTANCE is null
  THEN
    Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
                                     , :new.CREATED_IN_INSTANCE
                                     , :new.CREATED_DATE );
  END IF;
  ---
  ELSIF Updating
  THEN
    -- Set the MODIFIED... audit columns on row update.
    Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY
                                     , :new.MODIFIED_IN_INSTANCE
                                     , :new.MODIFIED_DATE );
    ---
  END IF;
END;
DECLARE
BEGIN
  -- Set the CREATED... audit columns on row insert if not supplied.
  IF :new.CREATED_BY is null or
     :new.CREATED_DATE is null or
     :new.CREATED_IN_INSTANCE is null
  THEN
    Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
                                     , :new.CREATED_IN_INSTANCE
                                     , :new.CREATED_DATE );
  END IF;
  ---
  ELSIF Updating
  THEN
    -- Set the MODIFIED... audit columns on row update.
    Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY

```

Oracle Designer Report : ckttrigger

Container: Real Property Management Version :

Table/View Name :II_RANGE_PERMITS_V (view)

```

, :new.MODIFIED_IN_INSTANCE
, :new.MODIFIED_DATE );

---
END IF;
END;
DECLARE
BEGIN
-- Set the CREATED... audit columns on row insert if not supplied.
IF :new.CREATED_BY is null or
:new.CREATED_DATE is null or
:new.CREATED_IN_INSTANCE is null
THEN
    Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
, :new.CREATED_IN_INSTANCE
, :new.CREATED_DATE );
END IF;
---
ELSIF Updating
THEN
-- Set the MODIFIED... audit columns on row update.
Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY
, :new.MODIFIED_IN_INSTANCE
, :new.MODIFIED_DATE );
---
END IF;
END;
DECLARE
BEGIN
-- Set the CREATED... audit columns on row insert if not supplied.
IF :new.CREATED_BY is null or
:new.CREATED_DATE is null or
:new.CREATED_IN_INSTANCE is null
THEN
    Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
, :new.CREATED_IN_INSTANCE
, :new.CREATED_DATE );
END IF;
---
ELSIF Updating
THEN
-- Set the MODIFIED... audit columns on row update.
Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY
, :new.MODIFIED_IN_INSTANCE
, :new.MODIFIED_DATE );
---
END IF;
END;
DECLARE
BEGIN
-- Set the CREATED... audit columns on row insert if not supplied.
IF :new.CREATED_BY is null or
:new.CREATED_DATE is null or
:new.CREATED_IN_INSTANCE is null
THEN
    Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
, :new.CREATED_IN_INSTANCE
, :new.CREATED_DATE );
END IF;

```

Oracle Designer Report : cktrigger

Container: Real Property Management Version :

Table/View Name :II_RANGE_PERMITS_V (view)

```

---
ELSIF Updating
THEN
  -- Set the MODIFIED... audit columns on row update.
  Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY
                                     , :new.MODIFIED_IN_INSTANCE
                                     , :new.MODIFIED_DATE );
---
END IF;
END;
-- PL/SQL Block
DECLARE
  --
  -- Cursor to check for existence of row in ii_range_permits
  cursor chs is
    select 'x'
      from ii_range_permits
     where accinst_cn = :old.cn;
  --
  eNullCN          exception;
  eNullID          exception;
  eNullObjTech     exception;
  eNullObjClass    exception;
  eNullObjName     exception;
  eNullEWFlag      exception;
  eNullBidBillFlag exception;
  eNullMngContCN   exception;
  --
  vAction varchar2(20) := null;
  vTable  varchar2(32) := null;
  vDummy  varchar2(1);
  --
  VRNGPAUDRecord ii_range_permit_audits%rowtype;
  --
BEGIN
  --
  -- Set audit column values
  VRNGPAUDRecord.changed_by      := user;
  VRNGPAUDRecord.changed_on     := sysdate;
  VRNGPAUDRecord.change_type    := 'U';
  VRNGPAUDRecord.old_id        := :old.ID;
  VRNGPAUDRecord.new_id        := :new.ID;
  VRNGPAUDRecord.old_cn        := :old.CN;
  VRNGPAUDRecord.new_cn        := :new.CN;
  VRNGPAUDRecord.old_description := :old.remarks;
  VRNGPAUDRecord.new_description := :new.remarks;
  VRNGPAUDRecord.old_managing_cont_cn := :old.managing_cont_cn;
  VRNGPAUDRecord.new_managing_cont_cn := :new.managing_cont_cn;
  VRNGPAUDRecord.old_obj_name    := :old.obj_name;
  VRNGPAUDRecord.new_obj_name    := :new.obj_name;
  VRNGPAUDRecord.old_permittee_cn := :old.permittee_cn;
  VRNGPAUDRecord.new_permittee_cn := :new.permittee_cn;
  VRNGPAUDRecord.old_supercedes_rngpmt_cn :=
:old.supercedes_rngpmt_cn;
  VRNGPAUDRecord.new_supercedes_rngpmt_cn :=
:new.supercedes_rngpmt_cn;
  VRNGPAUDRecord.old_split_bill_flag := :old.split_bill_flag;
  VRNGPAUDRecord.new_split_bill_flag := :new.split_bill_flag;

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :II_RANGE_PERMITS_V (view)

```

VRNGPAUDRecord.old_split_bill_month      := :old.split_bill_month;
VRNGPAUDRecord.new_split_bill_month      := :new.split_bill_month;
VRNGPAUDRecord.old_status                 := :old.status;
VRNGPAUDRecord.new_status                 := :new.status;
VRNGPAUDRecord.old_escrow_waiver_flag     :=
:old_escrow_waiver_flag;
VRNGPAUDRecord.new_escrow_waiver_flag     :=
:new_escrow_waiver_flag;
VRNGPAUDRecord.old_bid_bill_flag          := :old.bid_bill_flag;
VRNGPAUDRecord.new_bid_bill_flag          := :new.bid_bill_flag;
VRNGPAUDRecord.change_occ_in_table := 'II_RANGE_PERMITS_V';
--
if INSERTING then
  vAction := 'inserting';
  vTable := 'ACCOMPLISHMENT_INSTRUMENTS';
  if :new.cn is null then
    raise eNullCN;
  end if;
  if :new.id is null then
    raise eNullID;
  end if;
  if :new.obj_tech is null then
    raise eNullObjTech;
  end if;
  if :new.obj_class is null then
    raise eNullObjClass;
  end if;
  if :new.obj_name is null then
    raise eNullObjName;
  end if;
  if :new.managing_cont_cn is null then
    raise eNullMngContCN;
  end if;
  vTable := 'II_RANGE_PERMITS';
  if :new_escrow_waiver_flag is null then
    raise eNullEWFlag;
  end if;
  if :new_bid_bill_flag is null then
    raise eNullBidBillFlag;
  end if;
  --
  -- Insert row in ACCOMPLISHMENT_INSTRUMENTS
  vTable := 'ACCOMPLISHMENT_INSTRUMENTS';
  ii_range_permits_audit.VAuditCN :=
db_instance.generate_global_id('II_AUDTAB_CN_SEQ');
  insert into accomplishment_instruments(
    cn,
    id,
    description,
    managing_cont_cn,
    obj_tech,
    obj_class,
    obj_name,
    created_by,
    created_date,
    created_in_instance )
  values( :new.cn,
    :new.id,
    :new.remarks,

```

Oracle Designer Report : cktrigger

Container: Real Property Management Version :

Table/View Name :II_RANGE_PERMITS_V (view)

```

: new.managing_cont_cn,
: new.obj_tech,
: new.obj_class,
: new.obj_name,
user,
sysdate,
to_number(db_instance.get_pk_suffix)
);

--
-- Insert row into II_RANGE_PERMITS
vTable := 'II_RANGE_PERMITS';
insert into ii_range_permits ( accinst_cn,
                             status,
                             split_bill_flag,
                             split_bill_month,
                             permittee_cn,
                             supercedes_rngpmt_cn,
                             escrow_waiver_flag,
                             bid_bill_flag,
                             created_by,
                             created_date,
                             created_in_instance )
values( :new.cn,
        :new.status,
        :new.split_bill_flag,
        :new.split_bill_month,
        :new.permittee_cn,
        :new.supercedes_rngpmt_cn,
        :new.escrow_waiver_flag,
        :new.bid_bill_flag,
        user,
        sysdate,
        to_number(db_instance.get_pk_suffix) );

--
-- Insert row into audit table (II_RANGE_PERMIT_AUDITS)
VRNGPAUDRecord.changed_by      := user;
VRNGPAUDRecord.changed_on      := sysdate;
VRNGPAUDRecord.change_type     := 'I';
VRNGPAUDRecord.old_id          := null;
VRNGPAUDRecord.new_id          := :new.ID;
VRNGPAUDRecord.old_cn          := null;
VRNGPAUDRecord.new_cn          := :new.CN;
VRNGPAUDRecord.old_description := null;
VRNGPAUDRecord.new_description := :new.remarks;
VRNGPAUDRecord.old_managing_cont_cn := null;
VRNGPAUDRecord.new_managing_cont_cn :=
: new.managing_cont_cn;
VRNGPAUDRecord.old_obj_name     := null;
VRNGPAUDRecord.new_obj_name     := :new.obj_name;
VRNGPAUDRecord.old_permittee_cn := null;
VRNGPAUDRecord.new_permittee_cn := :new.permittee_cn;
VRNGPAUDRecord.old_supercedes_rngpmt_cn := null;
VRNGPAUDRecord.new_supercedes_rngpmt_cn :=
: new.supercedes_rngpmt_cn;
VRNGPAUDRecord.old_split_bill_flag := null;
VRNGPAUDRecord.new_split_bill_flag :=
: new.split_bill_flag;
VRNGPAUDRecord.old_split_bill_month := null;

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :II_RANGE_PERMITS_V (view)

```

        VRNGPAUDRecord.new_split_bill_month      :=
:new.split_bill_month;
        VRNGPAUDRecord.old_status                := null;
        VRNGPAUDRecord.new_status               := :new.status;
        VRNGPAUDRecord.old_escrow_waiver_flag    := null;
        VRNGPAUDRecord.new_escrow_waiver_flag    :=
:new_escrow_waiver_flag;
        VRNGPAUDRecord.old_bid_bill_flag        := null;
        VRNGPAUDRecord.new_bid_bill_flag        := :new.bid_bill_flag;
        VRNGPAUDRecord.change_occ_in_table := 'II_RANGE_PERMITS_V';
        ii_range_permits_audit.insert_II_RANGE_PERMIT_AUDITS (
VRNGPAUDRecord );
        ii_range_permits_audit.vAuditCN := null;
        --
    elsif UPDATING then
        vAction := 'updating';
        vTable := 'ACCOMPLISHMENT_INSTRUMENTS';
        if :new.cn is null then
            raise eNullCN;
        end if;
        if :new.id is null then
            raise eNullID;
        end if;
        if :new.obj_tech is null then
            raise eNullObjTech;
        end if;
        if :new.obj_class is null then
            raise eNullObjClass;
        end if;
        if :new.obj_name is null then
            raise eNullObjName;
        end if;
        if :new.managing_cont_cn is null then
            raise eNullMngContCN;
        end if;
        vTable := 'II_RANGE_PERMITS';
        if :new_escrow_waiver_flag is null then
            raise eNullEWFlag;
        end if;
        if :new_bid_bill_flag is null then
            raise eNullBidBillFlag;
        end if;
        --
        -- Update/insert II_RANGE_PERMITS row if something has changed
        ii_range_permits_audit.vAuditCN :=
db_instance.generate_global_id('II_AUDTAB_CN_SEQ');
        if (nvl(:new.status,'@')
nvl(:old.status,'@') or
            (nvl(:new.split_bill_flag,'@')
nvl(:old.split_bill_flag,'@') or
            (nvl(:new.split_bill_month,-1)
nvl(:old.split_bill_month,-1)) or
            (nvl(:new.bid_bill_flag,'@')
nvl(:old.bid_bill_flag,'@') or
            (nvl(:new_escrow_waiver_flag,'@')
nvl(:old_escrow_waiver_flag,'@') or
            (nvl(:new.permittee_cn,'@')
nvl(:old.permittee_cn,'@') or
            (nvl(:new.supercedes_rngpmt_cn,'@') !=

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :II_RANGE_PERMITS_V (view)

```

nvl(:old.supercedes_rngpmt_cn,'@')) then
  vTable := 'II_RANGE_PERMITS';
  open cHS;
  fetch cHS into vDummy;
  if cHS%notfound then
    close cHS;
    --
    -- Insert row into II_RANGE_PERMITS
    vTable := 'II_RANGE_PERMITS';
    vAction := 'inserting';
    insert into ii_range_permits (
      accinst_cn,
      status,
      split_bill_flag,
      split_bill_month,
      permittee_cn,
      supercedes_rngpmt_cn,
      escrow_waiver_flag,
      bid_bill_flag,
      created_by,
      created_date,
      created_in_instance )
      values( :old.cn,
        :new.status,
        :new.split_bill_flag,
        :new.split_bill_month,
        :new.permittee_cn,
        :new.supercedes_rngpmt_cn,
        :new.escrow_waiver_flag,
        :new.bid_bill_flag,
        user,
        sysdate,
        to_number(db_instance.get_pk_suffix) );
  else
    close cHS;
    --
    -- Update row in RANGE_PERMITS
    update ii_range_permits
      set status = :new.status,
        split_bill_flag = :new.split_bill_flag,
        split_bill_month = :new.split_bill_month,
        permittee_cn = :new.permittee_cn,
        supercedes_rngpmt_cn =
: new.supercedes_rngpmt_cn,
        escrow_waiver_flag = :new.escrow_waiver_flag,
        bid_bill_flag = :new.bid_bill_flag,
        modified_by = user,
        modified_date = sysdate,
        modified_in_instance =
to_number(db_instance.get_pk_suffix)
      where accinst_cn = :old.cn;
    end if;
  end if;
  --
  -- Update ACCPLISHMENT_INSTRUMENTS row is something has
  changed
  if (nvl(:new.id,'@') != nvl(:old.id,'@')) or
    (nvl(:new.remarks,'@') != nvl(:old.remarks,'@'))

```

Oracle Designer Report : cktrigger

Container: Real Property Management Version :

Table/View Name :II_RANGE_PERMITS_V (view)

```

or
    (nvl(:new.managing_cont_cn,'@') !=
nvl(:old.managing_cont_cn,'@')) then
    update accomplishment_instruments
        set id = :new.id,
            description = :new.remarks,
            managing_cont_cn = :new.managing_cont_cn,
            modified_by = user,
            modified_date = sysdate,
            modified_in_instance =
to_number(db_instance.get_pk_suffix)
        where cn = :old.cn;
    end if;
    --
    -- Insert row into audit table (II_RANGE_PERMIT_AUDITS)
    ii_range_permits_audit.Insert_II_RANGE_PERMIT_AUDITS (
VRNGPAUDRecord );
    ii_range_permits_audit.vAuditCN := null;
    --
    else
        vAction := 'deleting';
        ii_range_permits_audit.vAuditCN :=
db_instance.generate_global_id('II_AUDTAB_CN_SEQ');
        --
        -- Delete row from II_RANGE_PERMITS
        vTable := 'II_RANGE_PERMITS';
        delete from ii_range_permits
            where accinst_cn = :old.cn;
        --
        -- Delete row from ACCPLISHMENT_INSTRUMENTS
        vTable := 'ACCPLISHMENT_INSTRUMENTS';
        delete from accomplishment_instruments
            where cn = :old.cn;
        --
        -- Insert row into audit table (II_RANGE_PERMIT_AUDITS)
        VRNGPAUDRecord.changed_by := user;
        VRNGPAUDRecord.changed_on := sysdate;
        VRNGPAUDRecord.change_type := 'D';
        VRNGPAUDRecord.old_id := :old.ID;
        VRNGPAUDRecord.new_id := null;
        VRNGPAUDRecord.old_cn := :old.CN;
        VRNGPAUDRecord.new_cn := null;
        VRNGPAUDRecord.old_description := :old.remarks;
        VRNGPAUDRecord.new_description := null;
        VRNGPAUDRecord.old_managing_cont_cn :=
:old.managing_cont_cn;
        VRNGPAUDRecord.new_managing_cont_cn := null;
        VRNGPAUDRecord.old_obj_name := :old.obj_name;
        VRNGPAUDRecord.new_obj_name := null;
        VRNGPAUDRecord.old_permittee_cn := :old.permittee_cn;
        VRNGPAUDRecord.new_permittee_cn := null;
        VRNGPAUDRecord.old_supercedes_rngpmt_cn :=
:old.supercedes_rngpmt_cn;
        VRNGPAUDRecord.new_supercedes_rngpmt_cn := null;
        VRNGPAUDRecord.old_split_bill_flag :=
:old.split_bill_flag;
        VRNGPAUDRecord.new_split_bill_flag := null;
        VRNGPAUDRecord.old_split_bill_month :=
:old.split_bill_month;

```

Oracle Designer Report : ckttrigger

Container: Real Property Management Version :

Table/View Name :II_RANGE_PERMITS_V (view)

```

        VRNGPAUDRecord.new_split_bill_month      := null;
        VRNGPAUDRecord.old_status                := :old.status;
        VRNGPAUDRecord.new_status                := null;
        VRNGPAUDRecord.old_escrow_waiver_flag    :=
:old_escrow_waiver_flag;
        VRNGPAUDRecord.new_escrow_waiver_flag    := null;
        VRNGPAUDRecord.old_bid_bill_flag        := :old.bid_bill_flag;
        VRNGPAUDRecord.new_bid_bill_flag        := null;
        VRNGPAUDRecord.change_occ_in_table := 'II_RANGE_PERMITS_V';
        ii_range_permits_audit.Insert_II_RANGE_PERMIT_AUDITS (
VRNGPAUDRecord );
        ii_range_permits_audit.vAuditCN := null;
        --
    end if;
EXCEPTION
    when eNullID then
        raise_application_error(-20001,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Permit ID cannot be null');
    when eNullCN then
        raise_application_error(-20002,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Permit CN cannot be null');
    when eNullObjTech then
        raise_application_error(-20003,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Object Technology cannot be
null');
    when eNullObjClass then
        raise_application_error(-20004,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Object Class cannot be null');
    when eNullObjName then
        raise_application_error(-20005,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Object Name cannot be null');
    when eNullMngContCN then
        raise_application_error(-20006,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Managing/Issuing Organization
CN cannot be null');
    when eNullEWFlag then
        raise_application_error(-20007,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Escrow Waiver flag cannot be
null');
    when eNullBidBillFlag then
        raise_application_error(-20008,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Bid-Bill flag cannot be
null');
    when OTHERS then
        raise_application_error(-20009,'Error while '||vAction||'
'||vTable||' Row: '||
                                sqlerrm);
END;
```

Oracle Designer Report : ckttrigger

Container: Real Property Management

Version :

Table/View Name :LAND_UNITS (table)

```

Trigger Name : LAND_UNITS_TRG                      Create? Yes
PL/SQL Module
Name          : LAND_UNITS_TRG                      Enabled? Yes

PL/SQL File :

Purpose       : .

Trigger Time: BEFORE                               Level: ROW                      Insert? Yes
                                                    Delete? No
New Alias     NEW                                  Update? Yes
Old Alias     : OLD

```

Trigger Logic

```

-- PL/SQL Block
DECLARE
--
-- Cursor to check for existence of row in ii_range_permits
cursor chs is
  select 'x'
    from ii_range_permits
   where accinst_cn = :old.cn;
--
eNullCN      exception;
eNullID      exception;
eNullObjTech exception;
eNullObjClass exception;
eNullObjName exception;
eNullEWFlag  exception;
eNullBidBillFlag exception;
eNullMngContCN exception;
--
vAction varchar2(20) := null;
vTable  varchar2(32) := null;
vDummy  varchar2(1);
--
VRNGPAUDRecord ii_range_permit_audits%rowtype;
--
BEGIN
--
-- Set audit column values
VRNGPAUDRecord.changed_by      := user;
VRNGPAUDRecord.changed_on     := sysdate;
VRNGPAUDRecord.change_type    := 'U';
VRNGPAUDRecord.old_id         := :old.ID;
VRNGPAUDRecord.new_id         := :new.ID;
VRNGPAUDRecord.old_cn         := :old.CN;
VRNGPAUDRecord.new_cn         := :new.CN;
VRNGPAUDRecord.old_description := :old.remarks;
VRNGPAUDRecord.new_description := :new.remarks;
VRNGPAUDRecord.old_managing_cont_cn := :old.managing_cont_cn;
VRNGPAUDRecord.new_managing_cont_cn := :new.managing_cont_cn;
VRNGPAUDRecord.old_obj_name    := :old.obj_name;
VRNGPAUDRecord.new_obj_name    := :new.obj_name;
VRNGPAUDRecord.old_permittee_cn := :old.permittee_cn;
VRNGPAUDRecord.new_permittee_cn := :new.permittee_cn;

```

Oracle Designer Report : cktrigger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

    VRNGPAUDRecord.old_supercedes_rngpmt_cn :=
:old.supercedes_rngpmt_cn;
    VRNGPAUDRecord.new_supercedes_rngpmt_cn :=
:new.supercedes_rngpmt_cn;
    VRNGPAUDRecord.old_split_bill_flag      := :old.split_bill_flag;
    VRNGPAUDRecord.new_split_bill_flag      := :new.split_bill_flag;
    VRNGPAUDRecord.old_split_bill_month     := :old.split_bill_month;
    VRNGPAUDRecord.new_split_bill_month     := :new.split_bill_month;
    VRNGPAUDRecord.old_status               := :old.status;
    VRNGPAUDRecord.new_status               := :new.status;
    VRNGPAUDRecord.old_escrow_waiver_flag   :=
:old_escrow_waiver_flag;
    VRNGPAUDRecord.new_escrow_waiver_flag   :=
:new_escrow_waiver_flag;
    VRNGPAUDRecord.old_bid_bill_flag        := :old.bid_bill_flag;
    VRNGPAUDRecord.new_bid_bill_flag        := :new.bid_bill_flag;
    VRNGPAUDRecord.change_occ_in_table := 'II_RANGE_PERMITS_V';
--
if INSERTING then
    vAction := 'inserting';
    vTable := 'ACCOMPLISHMENT_INSTRUMENTS';
    if :new.cn is null then
        raise eNullCN;
    end if;
    if :new.id is null then
        raise eNullID;
    end if;
    if :new.obj_tech is null then
        raise eNullObjTech;
    end if;
    if :new.obj_class is null then
        raise eNullObjClass;
    end if;
    if :new.obj_name is null then
        raise eNullObjName;
    end if;
    if :new.managing_cont_cn is null then
        raise eNullMngContCN;
    end if;
    vTable := 'II_RANGE_PERMITS';
    if :new_escrow_waiver_flag is null then
        raise eNullEWFlag;
    end if;
    if :new_bid_bill_flag is null then
        raise eNullBidBillFlag;
    end if;
--
-- Insert row in ACCOMPLISHMENT_INSTRUMENTS
vTable := 'ACCOMPLISHMENT_INSTRUMENTS';
ii_range_permits_audit.vAuditCN :=
db_instance.generate_global_id('II_AUDTAB_CN_SEQ');
insert into accomplishment_instruments(
    cn,
    id,
    description,
    managing_cont_cn,
    obj_tech,
    obj_class,
    obj_name,

```

Oracle Designer Report : cktrigger

Container: Real Property Management

Version :

Table/View Name :LAND_UNITS (table)

```

        created_by,
        created_date,
        created_in_instance )
values( :new.cn,
        :new.id,
        :new.remarks,
        :new.managing_cont_cn,
        :new.obj_tech,
        :new.obj_class,
        :new.obj_name,
        user,
        sysdate,
        to_number(db_instance.get_pk_suffix)
);

--
-- Insert row into II_RANGE PERMITS
vTable := 'II_RANGE PERMITS';
insert into ii_range_permits ( accinst_cn,
                              status,
                              split_bill_flag,
                              split_bill_month,
                              permittee_cn,
                              supercedes_rngpmt_cn,
                              escrow_waiver_flag,
                              bid_bill_flag,
                              created_by,
                              created_date,
                              created_in_instance )
values( :new.cn,
        :new.status,
        :new.split_bill_flag,
        :new.split_bill_month,
        :new.permittee_cn,
        :new.supercedes_rngpmt_cn,
        :new.escrow_waiver_flag,
        :new.bid_bill_flag,
        user,
        sysdate,
        to_number(db_instance.get_pk_suffix) );

--
-- Insert row into audit table (II_RANGE PERMIT AUDITS)
VRNGPAUDRecord.changed_by      := user;
VRNGPAUDRecord.changed_on     := sysdate;
VRNGPAUDRecord.change_type    := 'I';
VRNGPAUDRecord.old_id        := null;
VRNGPAUDRecord.new_id        := :new.ID;
VRNGPAUDRecord.old_cn        := null;
VRNGPAUDRecord.new_cn        := :new.CN;
VRNGPAUDRecord.old_description := null;
VRNGPAUDRecord.new_description := :new.remarks;
VRNGPAUDRecord.old_managing_cont_cn := null;
VRNGPAUDRecord.new_managing_cont_cn :=
:new.managing_cont_cn;
VRNGPAUDRecord.old_obj_name   := null;
VRNGPAUDRecord.new_obj_name   := :new.obj_name;
VRNGPAUDRecord.old_permittee_cn := null;
VRNGPAUDRecord.new_permittee_cn := :new.permittee_cn;
VRNGPAUDRecord.old_supercedes_rngpmt_cn := null;

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

        VRNGPAUDRecord.new_supercedes_rngpmt_cn :=
:new.supercedes_rngpmt_cn;
VRNGPAUDRecord.old_split_bill_flag      := null;
VRNGPAUDRecord.new_split_bill_flag      :=
:new.split_bill_flag;
VRNGPAUDRecord.old_split_bill_month     := null;
VRNGPAUDRecord.new_split_bill_month     :=
:new.split_bill_month;
VRNGPAUDRecord.old_status               := null;
VRNGPAUDRecord.new_status               := :new.status;
VRNGPAUDRecord.old_escrow_waiver_flag   := null;
VRNGPAUDRecord.new_escrow_waiver_flag   :=
:new_escrow_waiver_flag;
VRNGPAUDRecord.old_bid_bill_flag        := null;
VRNGPAUDRecord.new_bid_bill_flag        := :new.bid_bill_flag;
VRNGPAUDRecord.change_occ_in_table := 'II RANGE PERMITS_V';
ii_range_permits_audit.insert_II_RANGE_PERMIT_AUDITS (
VRNGPAUDRecord );
ii_range_permits_audit.vAuditCN := null;
--
elseif UPDATING then
vAction := 'updating';
vTable := 'ACCOMPLISHMENT_INSTRUMENTS';
if :new.cn is null then
    raise eNullCN;
end if;
if :new.id is null then
    raise eNullID;
end if;
if :new.obj_tech is null then
    raise eNullObjTech;
end if;
if :new.obj_class is null then
    raise eNullObjClass;
end if;
if :new.obj_name is null then
    raise eNullObjName;
end if;
if :new.managing_cont_cn is null then
    raise eNullMngContCN;
end if;
vTable := 'II_RANGE_PERMITS';
if :new_escrow_waiver_flag is null then
    raise eNullEWFlag;
end if;
if :new.bid_bill_flag is null then
    raise eNullBidBillFlag;
end if;
--
-- Update/insert II_RANGE_PERMITS row if something has changed
ii_range_permits_audit.vAuditCN :=
db_instance.generate_global_id('II_AUDTAB_CN_SEQ');
if (nvl(:new.status,'@')
nvl(:old.status,'@')) or
    (nvl(:new.split_bill_flag,'@')
nvl(:old.split_bill_flag,'@')) or
    (nvl(:new.split_bill_month,-1)
nvl(:old.split_bill_month,-1)) or
    (nvl(:new.bid_bill_flag,'@')

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

nvl(:old.bid_bill_flag,'@')) or
    (nvl(:new.escrow_waiver_flag,'@') !=
nvl(:old.escrow_waiver_flag,'@')) or
    (nvl(:new.permittee_cn,'@') !=
nvl(:old.permittee_cn,'@')) or
    (nvl(:new.supercedes_rngpmt_cn,'@') !=
nvl(:old.supercedes_rngpmt_cn,'@')) then
    vTable := 'II_RANGE_PERMITS';
    open cHS;
    fetch cHS into vDummy;
    if cHS%notfound then
        close cHS;
        --
        -- Insert row into II RANGE PERMITS
        vTable := 'II_RANGE_PERMITS';
        vAction := 'inserting';
        insert into ii_range_permits (
            accinst_cn,
            status,
            split_bill_flag,
            split_bill_month,
            permittee_cn,
            supercedes_rngpmt_cn,
            escrow_waiver_flag,
            bid_bill_flag,
            created_by,
            created_date,
            created_in_instance )
            values( :old.cn,
                :new.status,
                :new.split_bill_flag,
                :new.split_bill_month,
                :new.permittee_cn,
                :new.supercedes_rngpmt_cn,
                :new.escrow_waiver_flag,
                :new.bid_bill_flag,
                user,
                sysdate,
                to_number(db_instance.get_pk_suffix) );
    else
        close cHS;
        --
        -- Update row in RANGE PERMITS
        update ii_range_permits
            set status = :new.status,
                split_bill_flag = :new.split_bill_flag,
                split_bill_month = :new.split_bill_month,
                permittee_cn = :new.permittee_cn,
                supercedes_rngpmt_cn =
: new.supercedes_rngpmt_cn,
                escrow_waiver_flag = :new.escrow_waiver_flag,
                bid_bill_flag = :new.bid_bill_flag,
                modified_by = user,
                modified_date = sysdate,
                modified_in_instance =
to_number(db_instance.get_pk_suffix)
            where accinst_cn = :old.cn;
    end if;

```

Oracle Designer Report : cktrigger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

end if;
--
-- Update ACCPLISHMENT_INSTRUMENTS row is something has
changed
if (nvl(:new.id, '@') != nvl(:old.id, '@')) or
   (nvl(:new.remarks, '@') != nvl(:old.remarks, '@'))
or
   (nvl(:new.managing_cont_cn, '@') !=
nvl(:old.managing_cont_cn, '@')) then
  update accomplishment_instruments
  set id = :new.id,
      description = :new.remarks,
      managing_cont_cn = :new.managing_cont_cn,
      modified_by = user,
      modified_date = sysdate,
      modified_in_instance =
to_number(db_instance.get_pk_suffix)
  where cn = :old.cn;
end if;
--
-- Insert row into audit table (II_RANGE_PERMIT_AUDITS)
ii_range_permits_audit.Insert_II_RANGE_PERMIT_AUDITS (
VRNGPAUDRecord);
ii_range_permits_audit.vAuditCN := null;
--
else
  vAction := 'deleting';
  ii_range_permits_audit.vAuditCN :=
db_instance.generate_global_id('II_AUDTAB_CN_SEQ');
--
-- Delete row from II_RANGE_PERMITS
vTable := 'II_RANGE_PERMITS';
delete from ii_range_permits
  where accinst_cn = :old.cn;
--
-- Delete row from ACCPLISHMENT_INSTRUMENTS
vTable := 'ACCPLISHMENT_INSTRUMENTS';
delete from accomplishment_instruments
  where cn = :old.cn;
--
-- Insert row into audit table (II_RANGE_PERMIT_AUDITS)
VRNGPAUDRecord.changed_by := user;
VRNGPAUDRecord.changed_on := sysdate;
VRNGPAUDRecord.change_type := 'D';
VRNGPAUDRecord.old_id := :old.ID;
VRNGPAUDRecord.new_id := null;
VRNGPAUDRecord.old_cn := :old.CN;
VRNGPAUDRecord.new_cn := null;
VRNGPAUDRecord.old_description := :old.remarks;
VRNGPAUDRecord.new_description := null;
VRNGPAUDRecord.old_managing_cont_cn :=
:old.managing_cont_cn;
VRNGPAUDRecord.new_managing_cont_cn := null;
VRNGPAUDRecord.old_obj_name := :old.obj_name;
VRNGPAUDRecord.new_obj_name := null;
VRNGPAUDRecord.old_permittee_cn := :old.permittee_cn;
VRNGPAUDRecord.new_permittee_cn := null;
VRNGPAUDRecord.old_supercedes_rngpmt_cn :=
:old.supercedes_rngpmt_cn;

```

Oracle Designer Report : cktggr

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

        VRNGPAUDRecord.new_supercedes_rngpmt_cn := null;
        VRNGPAUDRecord.old_split_bill_flag      :=
:old.split_bill_flag;
        VRNGPAUDRecord.new_split_bill_flag      := null;
        VRNGPAUDRecord.old_split_bill_month     :=
:old.split_bill_month;
        VRNGPAUDRecord.new_split_bill_month     := null;
        VRNGPAUDRecord.old_status               := :old.status;
        VRNGPAUDRecord.new_status               := null;
        VRNGPAUDRecord.old_escrow_waiver_flag   :=
:old_escrow_waiver_flag;
        VRNGPAUDRecord.new_escrow_waiver_flag   := null;
        VRNGPAUDRecord.old_bid_bill_flag        := :old.bid_bill_flag;
        VRNGPAUDRecord.new_bid_bill_flag        := null;
        VRNGPAUDRecord.change_occ_in_table := 'II_RANGE_PERMITS_V';
        ii_range_permits_audit.insert_II_RANGE_PERMIT_AUDITS (
VRNGPAUDRecord );
        ii_range_permits_audit.vAuditCN := null;
        --
    end if;
EXCEPTION
    when eNullID then
        raise_application_error(-20001,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Permit ID cannot be null');
    when eNullCN then
        raise_application_error(-20002,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Permit CN cannot be null');
    when eNullObjTech then
        raise_application_error(-20003,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Object Technology cannot be
null');
    when eNullObjClass then
        raise_application_error(-20004,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Object Class cannot be null');
    when eNullObjName then
        raise_application_error(-20005,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Object Name cannot be null');
    when eNullMngContCN then
        raise_application_error(-20006,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Managing/Issuing Organization
CN cannot be null');
    when eNullEWFlag then
        raise_application_error(-20007,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Escrow Waiver flag cannot be
null');
    when eNullBidBillFlag then
        raise_application_error(-20008,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Bid-Bill flag cannot be
null');
    when OTHERS then
        raise_application_error(-20009,'Error while '||vAction||'

```

Oracle Designer Report : cktrigger

Container: Real Property Management

Version :

Table/View Name :LAND_UNITS (table)

```

'||vTable||' row: '||
                                sqlerrm);

END;
-- PL/SQL Block
DECLARE
--
-- Cursor to check for existence of row in ii_range_permits
cursor cHS is
  select 'x'
    from ii_range_permits
   where accinst_cn = :old.cn;
--
eNullCN          exception;
eNullID          exception;
eNullObjTech     exception;
eNullObjClass    exception;
eNullObjName     exception;
eNullEWFlag      exception;
eNullBidBillFlag exception;
eNullMngContCN   exception;
--
vAction varchar2(20) := null;
vTable  varchar2(32) := null;
vDummy  varchar2(1);
--
VRNGPAUDRecord ii_range_permit_audits%rowtype;
--
BEGIN
--
-- Set audit column values
VRNGPAUDRecord.changed_by      := user;
VRNGPAUDRecord.changed_on     := sysdate;
VRNGPAUDRecord.change_type    := 'U';
VRNGPAUDRecord.old_id         := :old.ID;
VRNGPAUDRecord.new_id         := :new.ID;
VRNGPAUDRecord.old_cn         := :old.CN;
VRNGPAUDRecord.new_cn         := :new.CN;
VRNGPAUDRecord.old_description := :old.remarks;
VRNGPAUDRecord.new_description := :new.remarks;
VRNGPAUDRecord.old_managing_cont_cn := :old.managing_cont_cn;
VRNGPAUDRecord.new_managing_cont_cn := :new.managing_cont_cn;
VRNGPAUDRecord.old_obj_name    := :old.obj_name;
VRNGPAUDRecord.new_obj_name    := :new.obj_name;
VRNGPAUDRecord.old_permittee_cn := :old.permittee_cn;
VRNGPAUDRecord.new_permittee_cn := :new.permittee_cn;
VRNGPAUDRecord.old_supercedes_rngpmt_cn :=
:old.supercedes_rngpmt_cn;
VRNGPAUDRecord.new_supercedes_rngpmt_cn :=
:new.supercedes_rngpmt_cn;
VRNGPAUDRecord.old_split_bill_flag := :old.split_bill_flag;
VRNGPAUDRecord.new_split_bill_flag := :new.split_bill_flag;
VRNGPAUDRecord.old_split_bill_month := :old.split_bill_month;
VRNGPAUDRecord.new_split_bill_month := :new.split_bill_month;
VRNGPAUDRecord.old_status           := :old.status;
VRNGPAUDRecord.new_status           := :new.status;
VRNGPAUDRecord.old_escrow_waiver_flag :=
:old_escrow_waiver_flag;
VRNGPAUDRecord.new_escrow_waiver_flag :=

```

Oracle Designer Report : ckttrigger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

:new.escrow_waiver_flag;
VRNGPAUDRecord.old_bid_bill_flag      := :old.bid_bill_flag;
VRNGPAUDRecord.new_bid_bill_flag      := :new.bid_bill_flag;
VRNGPAUDRecord.change_occ_in_table := 'II_RANGE_PERMITS_V';
--
if INSERTING then
  vAction := 'inserting';
  vTable := 'ACCOMPLISHMENT_INSTRUMENTS';
  if :new.cn is null then
    raise eNullCN;
  end if;
  if :new.id is null then
    raise eNullID;
  end if;
  if :new.obj_tech is null then
    raise eNullObjTech;
  end if;
  if :new.obj_class is null then
    raise eNullObjClass;
  end if;
  if :new.obj_name is null then
    raise eNullObjName;
  end if;
  if :new.managing_cont_cn is null then
    raise eNullMngContCN;
  end if;
  vTable := 'II_RANGE_PERMITS';
  if :new.escrow_waiver_flag is null then
    raise eNullEWFlag;
  end if;
  if :new.bid_bill_flag is null then
    raise eNullBidBillFlag;
  end if;
  --
  -- Insert row in ACCOMPLISHMENT_INSTRUMENTS
  vTable := 'ACCOMPLISHMENT_INSTRUMENTS';
  ii_range_permits_audit.VAuditCN :=
db_instance.generate_global_id('II_AUDTAB_CN_SEQ');
  insert into accomplishment_instruments(
    cn,
    id,
    description,
    managing_cont_cn,
    obj_tech,
    obj_class,
    obj_name,
    created_by,
    created_date,
    created_in_instance )
  values( :new.cn,
    :new.id,
    :new.remarks,
    :new.managing_cont_cn,
    :new.obj_tech,
    :new.obj_class,
    :new.obj_name,
    user,
    sysdate,
    to_number(db_instance.get_pk_suffix)

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```
);
--
-- Insert row into II_RANGE_PERMITS
vTable := 'II_RANGE_PERMITS';
insert into ii_range_permits ( accinst_cn,
                             status,
                             split_bill_flag,
                             split_bill_month,
                             permittee_cn,
                             supercedes_rngpmt_cn,
                             escrow_waiver_flag,
                             bid_bill_flag,
                             created_by,
                             created_date,
                             created_in_instance )
values( :new.cn,
        :new.status,
        :new.split_bill_flag,
        :new.split_bill_month,
        :new.permittee_cn,
        :new.supercedes_rngpmt_cn,
        :new.escrow_waiver_flag,
        :new.bid_bill_flag,
        user,
        sysdate,
        to_number(db_instance.get_pk_suffix) );
--
-- Insert row into audit table (II_RANGE_PERMIT_AUDITS)
VRNGPAUDRecord.changed_by      := user;
VRNGPAUDRecord.changed_on      := sysdate;
VRNGPAUDRecord.change_type     := 'I';
VRNGPAUDRecord.old_id          := null;
VRNGPAUDRecord.new_id          := :new.ID;
VRNGPAUDRecord.old_cn         := null;
VRNGPAUDRecord.new_cn          := :new.CN;
VRNGPAUDRecord.old_description := null;
VRNGPAUDRecord.new_description := :new.remarks;
VRNGPAUDRecord.old_managing_cont_cn := null;
VRNGPAUDRecord.new_managing_cont_cn :=
:new.managing_cont_cn;
VRNGPAUDRecord.old_obj_name    := null;
VRNGPAUDRecord.new_obj_name    := :new.obj_name;
VRNGPAUDRecord.old_permittee_cn := null;
VRNGPAUDRecord.new_permittee_cn := :new.permittee_cn;
VRNGPAUDRecord.old_supercedes_rngpmt_cn := null;
VRNGPAUDRecord.new_supercedes_rngpmt_cn :=
:new.supercedes_rngpmt_cn;
VRNGPAUDRecord.old_split_bill_flag := null;
VRNGPAUDRecord.new_split_bill_flag :=
:new.split_bill_flag;
VRNGPAUDRecord.old_split_bill_month := null;
VRNGPAUDRecord.new_split_bill_month :=
:new.split_bill_month;
VRNGPAUDRecord.old_status          := null;
VRNGPAUDRecord.new_status          := :new.status;
VRNGPAUDRecord.old_escrow_waiver_flag := null;
VRNGPAUDRecord.new_escrow_waiver_flag :=
:new.escrow_waiver_flag;
```

Oracle Designer Report : cktrigger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

VRNGPAUDRecord.old_bid_bill_flag      := null;
VRNGPAUDRecord.new_bid_bill_flag      := :new.bid_bill_flag;
VRNGPAUDRecord.change_occ_in_table := 'II RANGE PERMITS V';
ii_range_permits_audit.insert_ii_range_permit_audits (
VRNGPAUDRecord );
ii_range_permits_audit.vAuditCN := null;
--
elseif UPDATING then
vAction := 'updating';
vTable := 'ACCOMPLISHMENT_INSTRUMENTS';
if :new.cn is null then
raise eNullCN;
end if;
if :new.id is null then
raise eNullID;
end if;
if :new.obj_tech is null then
raise eNullObjTech;
end if;
if :new.obj_class is null then
raise eNullObjClass;
end if;
if :new.obj_name is null then
raise eNullObjName;
end if;
if :new.managing_cont_cn is null then
raise eNullMngContCN;
end if;
vTable := 'II_RANGE_PERMITS';
if :new.escrow_waiver_flag is null then
raise eNullEWFlag;
end if;
if :new.bid_bill_flag is null then
raise eNullBidBillFlag;
end if;
--
-- Update/insert II_RANGE_PERMITS row if something has changed
ii_range_permits_audit.vAuditCN :=
db_instance.generate_global_id('II_AUDTAB_CN_SEQ');
if (nvl(:new.status,'@')
nvl(:old.status,'@')) or
(nvl(:new.split_bill_flag,'@')
nvl(:old.split_bill_flag,'@')) or
(nvl(:new.split_bill_month,-1)
nvl(:old.split_bill_month,-1)) or
(nvl(:new.bid_bill_flag,'@')
nvl(:old.bid_bill_flag,'@')) or
(nvl(:new.escrow_waiver_flag,'@')
nvl(:old.escrow_waiver_flag,'@')) or
(nvl(:new.permittee_cn,'@')
nvl(:old.permittee_cn,'@')) or
(nvl(:new.supercedes_rngpmt_cn,'@')
nvl(:old.supercedes_rngpmt_cn,'@')) then
vTable := 'II_RANGE_PERMITS';
open cHS;
fetch cHS into vDummy;
if cHS%notfound then
close cHS;
--

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

-- Insert row into II_RANGE_PERMITS
vTable := 'II_RANGE_PERMITS';
vAction := 'inserting';
insert into ii_range_permits (
    accinst_cn,
    status,
    split_bill_flag,
    split_bill_month,
    permittee_cn,
    supercedes_rngpmt_cn,
    escrow_waiver_flag,
    bid_bill_flag,
    created_by,
    created_date,
    created_in_instance )
values( :old.cn,
        :new.status,
        :new.split_bill_flag,
        :new.split_bill_month,
        :new.permittee_cn,
        :new.supercedes_rngpmt_cn,
        :new.escrow_waiver_flag,
        :new.bid_bill_flag,
        user,
        sysdate,
        :new.created_in_instance );

to_number(db_instance.get_pk_suffix) );
else
    close cHS;
--
-- Update row in RANGE PERMITS
update ii_range_permits
set status = :new.status,
    split_bill_flag = :new.split_bill_flag,
    split_bill_month = :new.split_bill_month,
    permittee_cn = :new.permittee_cn,
    supercedes_rngpmt_cn =
:new.supercedes_rngpmt_cn,
    escrow_waiver_flag = :new.escrow_waiver_flag,
    bid_bill_flag = :new.bid_bill_flag,
    modified_by = user,
    modified_date = sysdate,
    modified_in_instance =
to_number(db_instance.get_pk_suffix)
    where accinst_cn = :old.cn;
end if;
end if;
--
-- Update ACCPLISHMENT_INSTRUMENTS row is something has
changed
if (nvl(:new.id, '@') != nvl(:old.id, '@')) or
   (nvl(:new.remarks, '@') != nvl(:old.remarks, '@'))
or
   (nvl(:new.managing_cont_cn, '@') !=
nvl(:old.managing_cont_cn, '@')) then
    update accomplishment_instruments
    set id = :new.id,
        description = :new.remarks,
        managing_cont_cn = :new.managing_cont_cn,

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

        modified by = user,
        modified_date = sysdate,
        modified in instance =
to_number(db_instance.get_pk_suffix)
        where cn = :old.cn;
    end if;
    --
    -- Insert row into audit table (II_RANGE_PERMIT_AUDITS)
    ii_range_permits_audit.Insert_II_RANGE_PERMIT_AUDITS (
VRNGPAUDRecord );
    ii_range_permits_audit.vAuditCN := null;
    --
    else
        vAction := 'deleting';
        ii_range_permits_audit.vAuditCN :=
db_instance.generate_global_id('II_AUDTAB_CN_SEQ');
    --
    -- Delete row from II_RANGE_PERMITS
    vTable := 'II_RANGE_PERMITS';
    delete from ii_range_permits
        where accinst_cn = :old.cn;
    --
    -- Delete row from ACCPLISHMENT INSTRUMENTS
    vTable := 'ACCPLISHMENT INSTRUMENTS';
    delete from accomplishment_instruments
        where cn = :old.cn;
    --
    -- Insert row into audit table (II_RANGE_PERMIT_AUDITS)
    VRNGPAUDRecord.changed_by           := user;
    VRNGPAUDRecord.changed_on           := sysdate;
    VRNGPAUDRecord.change_type          := 'D';
    VRNGPAUDRecord.old_id               := :old.ID;
    VRNGPAUDRecord.new_id               := null;
    VRNGPAUDRecord.old_cn               := :old.CN;
    VRNGPAUDRecord.new_cn               := null;
    VRNGPAUDRecord.old_description      := :old.remarks;
    VRNGPAUDRecord.new_description      := null;
    VRNGPAUDRecord.old_managing_cont_cn :=
:old.managing_cont_cn;
    VRNGPAUDRecord.new_managing_cont_cn := null;
    VRNGPAUDRecord.old_obj_name         := :old.obj_name;
    VRNGPAUDRecord.new_obj_name         := null;
    VRNGPAUDRecord.old_permittee_cn     := :old.permittee_cn;
    VRNGPAUDRecord.new_permittee_cn     := null;
    VRNGPAUDRecord.old_supercedes_rngpmt_cn :=
:old.supercedes_rngpmt_cn;
    VRNGPAUDRecord.new_supercedes_rngpmt_cn := null;
    VRNGPAUDRecord.old_split_bill_flag :=
:old.split_bill_flag;
    VRNGPAUDRecord.new_split_bill_flag  := null;
    VRNGPAUDRecord.old_split_bill_month :=
:old.split_bill_month;
    VRNGPAUDRecord.new_split_bill_month := null;
    VRNGPAUDRecord.old_status           := :old.status;
    VRNGPAUDRecord.new_status           := null;
    VRNGPAUDRecord.old_escrow_waiver_flag :=
:old_escrow_waiver_flag;
    VRNGPAUDRecord.new_escrow_waiver_flag := null;
    VRNGPAUDRecord.old_bid_bill_flag    := :old.bid_bill_flag;

```

Oracle Designer Report : cktrigger

Container: Real Property Management

Version :

Table/View Name :LAND_UNITS (table)

```

        VRNGPAUDRecord.new_bid_bill_flag          := null;
        VRNGPAUDRecord.change_occ_in table := 'II_RANGE PERMITS_V';
        ii_range_permits_audit.Insert_II_RANGE_PERMIT_AUDITS (
VRNGPAUDRecord );
        ii_range_permits_audit.vAuditCN := null;
        --
    end if;
EXCEPTION
    when eNullID then
        raise_application_error(-20001,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Permit ID cannot be null');
    when eNullCN then
        raise_application_error(-20002,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Permit CN cannot be null');
    when eNullObjTech then
        raise_application_error(-20003,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Object Technology cannot be
null');
    when eNullObjClass then
        raise_application_error(-20004,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Object Class cannot be null');
    when eNullObjName then
        raise_application_error(-20005,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Object Name cannot be null');
    when eNullMngContCN then
        raise_application_error(-20006,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Managing/Issuing Organization
CN cannot be null');
    when eNullEWFlag then
        raise_application_error(-20007,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Escrow Waiver flag cannot be
null');
    when eNullBidBillFlag then
        raise_application_error(-20008,'Error while '||vAction||'
'||vTable||' Row: '||
                                'Bid-Bill flag cannot be
null');
    when OTHERS then
        raise_application_error(-20009,'Error while '||vAction||'
'||vTable||' Row: '||
                                sqlerrm);
END;
DECLARE
BEGIN
    DECLARE
        cursor lu_cur is
            select root_cn
            from land_units
            where cn = :new.parent_cn;
            l_root_cn varchar2(34);
BEGIN

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

        IF INSERTING THEN
            IF not ii_chk_role_assign('INFRA SVC MGR') and
               ii_cent_mgd_lu_obj(:new.obj_name) then
                raise_application_error(-20304,'Insufficient privileges
to '||
               'insert '||initcap(replace(:new.obj_name,'_',' '))||'
Land Units');
            END IF;
            --
            -- Added by RHA for v4.7
            --
            IF :new.cn IS NULL THEN
                :new.cn := db_instance.generate_global_id('LU_CN_SEQ');
            END IF;
            db_instance.insert_audit_columns
            (:new.created_by, :new.created_in_instance,
:new.created_date);
            --
            --
            IF :new.parent_cn IS NULL THEN
                :new.root_cn := :new.cn;
            ELSE
                IF :new.root_cn IS NULL THEN
                    open lu_cur;
                    fetch lu_cur into l_root_cn;
                    close lu_cur;
                    :new.root_cn := l_root_cn;
                END IF;
            END IF;
        ELSE
            IF not ii_chk_role_assign('INFRA SVC MGR') and
               ii_cent_mgd_lu_obj(:new.obj_name) and
               :new.obj_name != :old.obj_name THEN
                raise_application_error(-20303,'Insufficient privileges
to '||
               'change a Land Unit from '||
               initcap(replace(:old.obj_name,'_',' '))||' to '||
               initcap(replace(:new.obj_name,'_',' '));
            END IF;
            IF not ii_chk_role_assign('INFRA SVC MGR') and
               ii_cent_mgd_lu_obj(:old.obj_name) then
                IF :old.obj_name = 'WILDERNESS' then
                    IF (:old.obj_tech != :new.obj_tech) or
                       (:old.obj_class != :new.obj_class) or
                       (:old.obj_name != :new.obj_name) or
                       (nvl(:old.id,'@') != nvl(:new.id,'@')) or
                       (nvl(:old.name,'@') != nvl(:new.name,'@')) then
                        raise_application_error(-20302,'Insufficient
privileges to '||
                        'update '||
initcap(replace(:old.obj_name,'_',' '))||
                        ' Land Unit columns');
                    END IF;
                ELSE
                    raise_application_error(-20301,'Insufficient
privileges to '||
                    'update '||
initcap(replace(:old.obj_name,'_',' '))||
                    ' Land Units');
                END IF;
            END IF;
        END IF;
    
```

Oracle Designer Report : cktrigger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

        END IF;
    END IF;
    db_instance.update_audit_columns
        (:new.modified_by, :new.modified_in_instance,
:new.modified_date);
    --
    -- Added by RAdams on 6/24/2002 for v5.0:
    --
    -- Record old/new ID and CN in PL/SQL table for purposes of
    -- writing row to range permits audit table
    (ii_range_pmt_li_audit) when the updated
    -- Land Unit ID is an RMU, State, Proclaimed Unit, or
    County referenced in a
    -- grazing permit. This action takes place in the
    statement-level
    -- after-update/delete trigger that first queries the range
    permit line items table
    -- to check if this land unit is an RMU, State, Proclaimed
    Unit, or County referenced in
    -- a grazing permit. If this is the case, a re-query on
    the land_units table is necessary
    -- to get the remaining RMU, State, Proclaimed Unit, and
    County IDs to be inserted
    -- into the audit table. To avoid the "mutating table"
    error, the re-query of the land_units
    -- table needed to be moved to a statement-level after
    update/delete trigger, passing "old" and "new"
    -- values in the package PL/SQL table referenced in the
    code below.
    --
    if nvl(:new.id,'@') != nvl(:old.id,'@') then
        ii_range_permits_audit.vNumEntries :=
ii_range_permits_audit.vNumEntries + 1;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ne
w_id := :new.id;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ol
d_id := :old.id;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ne
w_cn := :new.cn;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ol
d_cn := :old.cn;
        end if;
    --
    -- Need to add root_cn update logic. (RHA 3/4/98)
    END IF;
END;
END;
```

Container: Real Property Management

Version :

Table/View Name :LAND_UNITS (table)

Trigger Name :	LU_IICODE_TRG	Create?	Yes
PL/SQL Module			
Name :	LU_IICODE_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
New Alias	NEW	Delete?	No
Old Alias :	OLD	Update?	Yes

Trigger Logic

```

DECLARE
BEGIN
  -- Check Area UoM
  IF (:new.AREA_UOM IS NOT NULL
      AND NOT chk_iicode('AREA_UOM', :new.AREA_UOM))
  THEN
    raise_application_error(-20301,
      'Invalid value "'||:new.AREA_UOM||'" for Area Unit');
  END IF;
  -- Check Avg. Elevation Uom
  IF (:new.AVG_ELEVATION_UOM IS NOT NULL
      AND NOT chk_iicode('ELEVATION_UOM', :new.AVG_ELEVATION_UOM))
  THEN
    raise_application_error(-20302,
      'Invalid value "'||:new.AVG_ELEVATION_UOM||'" for
Elevation Unit');
  END IF;
  -- Check Predominant Landform
  IF (:new.PREDOMINANT_LANDFORM IS NOT NULL
      AND NOT chk_iicode('LU_LANDFORM', :new.PREDOMINANT_LANDFORM))
  THEN
    raise_application_error(-20303,
      'Invalid value "'||:new.PREDOMINANT_LANDFORM||'" for
Predominant '||
      'Landform');
  END IF;
  -- Check Predominant Surface Substrate
  IF (:new.PREDOMNT_SFC_SUBSTRATE IS NOT NULL
      AND NOT chk_iicode('LU_SURFACE_SUBSTRATE',
:new.PREDOMNT_SFC_SUBSTRATE))
  THEN
    raise_application_error(-20304,
      'Invalid value "'||:new.PREDOMNT_SFC_SUBSTRATE||'" for
Predominant '||
      'Surface Substrate');
  END IF;
  -- Check LU Subtype
  IF (:new.LU_SUBTYPE IS NOT NULL
      AND NOT chk_iicode(:new.OBJ_NAME||'_SUBTYPE', upper(
:new.LU_SUBTYPE)))
  THEN
    raise_application_error(-20305,

```

Oracle Designer Report : ckttrigger

Container: Real Property Management **Version :**

Table/View Name : LAND_UNITS (table)

```

                                'Invalid value "':||:new.LU_SUBTYPE||'" for '||
                                initcap(replace(:new.OBJ_NAME, '_',' '))||
                                ' Sub-Type');
        END IF;
    END;
```

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

Trigger Name : II_LU_RPA_AR_TRG Create? Yes
 PL/SQL Module
 Name : II_LU_RPA_AR_TRG Enabled? Yes
 PL/SQL File :
 Purpose : Maintain Ins/Upd/Del after each row for Real Property Accounting.
 Trigger Time: AFTER Level: ROW Insert? Yes
 Delete? Yes
 New Alias Update? Yes
 Old Alias :

Trigger Logic

```
DECLARE
BEGIN
  -- Check Area UoM
  IF (:new.AREA_UOM IS NOT NULL
    AND NOT chk_iicode('AREA_UOM', :new.AREA_UOM))
  THEN
    raise_application_error(-20301,
      'Invalid value "'||:new.AREA_UOM||'" for Area Unit');
  END IF;
  -- Check Avg. Elevation Uom
  IF (:new.AVG_ELEVATION_UOM IS NOT NULL
    AND NOT chk_iicode('ELEVATION_UOM', :new.AVG_ELEVATION_UOM))
  THEN
    raise_application_error(-20302,
      'Invalid value "'||:new.AVG_ELEVATION_UOM||'" for
Elevation Unit');
  END IF;
  -- Check Predominant Landform
  IF (:new.PREDOMINANT_LANDFORM IS NOT NULL
    AND NOT chk_iicode('LU_LANDFORM', :new.PREDOMINANT_LANDFORM))
  THEN
    raise_application_error(-20303,
      'Invalid value "'||:new.PREDOMINANT_LANDFORM||'" for
Predominant '||
      'Landform');
  END IF;
  -- Check Predominant Surface Substrate
  IF (:new.PREDOMNT_SFC_SUBSTRATE IS NOT NULL
    AND NOT chk_iicode('LU_SURFACE_SUBSTRATE',
: new.PREDOMNT_SFC_SUBSTRATE))
  THEN
    raise_application_error(-20304,
      'Invalid value "'||:new.PREDOMNT_SFC_SUBSTRATE||'" for
Predominant '||
      'Surface Substrate');
  END IF;
  -- Check LU Subtype
  IF (:new.LU_SUBTYPE IS NOT NULL
    AND NOT chk_iicode(:new.OBJ_NAME||'_SUBTYPE', upper(
: new.LU_SUBTYPE)))
  THEN
```

Oracle Designer Report : ckttrigger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

        raise_application_error(-20305,
            'Invalid value "'||:new.LU_SUBTYPE||'" for '||
            initcap(replace(:new.OBJ_NAME, '-', ' '))||
            ' Sub-Type');
    END IF;
END;
DECLARE
BEGIN
    -- Check Area UoM
    IF (:new.AREA_UOM IS NOT NULL
        AND NOT chk_iicode('AREA_UOM', :new.AREA_UOM))
    THEN
        raise_application_error(-20301,
            'Invalid value "'||:new.AREA_UOM||'" for Area Unit');
    END IF;
    -- Check Avg. Elevation Uom
    IF (:new.AVG_ELEVATION_UOM IS NOT NULL
        AND NOT chk_iicode('ELEVATION_UOM', :new.AVG_ELEVATION_UOM))
    THEN
        raise_application_error(-20302,
            'Invalid value "'||:new.AVG_ELEVATION_UOM||'" for
Elevation Unit');
    END IF;
    -- Check Predominant Landform
    IF (:new.PREDOMINANT_LANDFORM IS NOT NULL
        AND NOT chk_iicode('LU_LANDFORM', :new.PREDOMINANT_LANDFORM))
    THEN
        raise_application_error(-20303,
            'Invalid value "'||:new.PREDOMINANT_LANDFORM||'" for
Predominant '||
            'Landform');
    END IF;
    -- Check Predominant Surface Substrate
    IF (:new.PREDOMNT_SFC_SUBSTRATE IS NOT NULL
        AND NOT chk_iicode('LU_SURFACE_SUBSTRATE',
:new.PREDOMNT_SFC_SUBSTRATE))
    THEN
        raise_application_error(-20304,
            'Invalid value "'||:new.PREDOMNT_SFC_SUBSTRATE||'" for
Predominant '||
            'Surface Substrate');
    END IF;
    -- Check LU Subtype
    IF (:new.LU_SUBTYPE IS NOT NULL
        AND NOT chk_iicode(:new.OBJ_NAME||'_SUBTYPE', upper(
:new.LU_SUBTYPE)))
    THEN
        raise_application_error(-20305,
            'Invalid value "'||:new.LU_SUBTYPE||'" for '||
            initcap(replace(:new.OBJ_NAME, '-', ' '))||
            ' Sub-Type');
    END IF;
END;
DECLARE
BEGIN
    -- Check Area UoM
    IF (:new.AREA_UOM IS NOT NULL
        AND NOT chk_iicode('AREA_UOM', :new.AREA_UOM))

```

Oracle Designer Report : ckttrigger

Container: Real Property Management

Version :

Table/View Name :LAND_UNITS (table)

```

THEN
    raise_application_error(-20301,
        'Invalid value "'||:new.AREA_UOM||'" for Area Unit');
END IF;
-- Check Avg. Elevation Uom
IF (:new.AVG_ELEVATION_UOM IS NOT NULL
    AND NOT chk_iicode('ELEVATION_UOM', :new.AVG_ELEVATION_UOM))
THEN
    raise_application_error(-20302,
        'Invalid value "'||:new.AVG_ELEVATION_UOM||'" for
Elevation Unit');
END IF;
-- Check Predominant Landform
IF (:new.PREDOMINANT_LANDFORM IS NOT NULL
    AND NOT chk_iicode('LU_LANDFORM', :new.PREDOMINANT_LANDFORM))
THEN
    raise_application_error(-20303,
        'Invalid value "'||:new.PREDOMINANT_LANDFORM||'" for
Predominant '||
        'Landform');
END IF;
-- Check Predominant Surface Substrate
IF (:new.PREDOMNT_SFC_SUBSTRATE IS NOT NULL
    AND NOT chk_iicode('LU_SURFACE_SUBSTRATE',
:new.PREDOMNT_SFC_SUBSTRATE))
THEN
    raise_application_error(-20304,
        'Invalid value "'||:new.PREDOMNT_SFC_SUBSTRATE||'" for
Predominant '||
        'Surface Substrate');
END IF;
-- Check LU Subtype
IF (:new.LU_SUBTYPE IS NOT NULL
    AND NOT chk_iicode(:new.OBJ_NAME||'_SUBTYPE',upper(
:new.LU_SUBTYPE)))
THEN
    raise_application_error(-20305,
        'Invalid value "'||:new.LU_SUBTYPE||'" for '||
        initcap(replace(:new.OBJ_NAME,'_',' '))||
        ' Sub-Type');
END IF;
END;
DECLARE
    vModuleName CONSTANT VARCHAR2(30) := 'II_LU_RPA_AR_TRG';
Added on 01/07/2003
BEGIN
    -- DELETING
    IF DELETING THEN
        -- DELETE from ii_rpa_assets_v_tab
        DELETE FROM ii_rpa_assets_v_tab
        WHERE lu_cn = :old.cn;

    -- INSERTING
    ELSIF INSERTING THEN
        -- Insert into table ii_rpa_assets_v_tab
        INSERT INTO ii_rpa_assets_v_tab (
            FEA_CN
            ,LU_CN

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

,PLD_CN
,OBJECT_NAME
,OBJECT_CLASS
,ID
,NAME
,SUB_TYPE
,OWNERSHIP
,ADMIN_ORG
,DEVELOPMENT_STATUS
,CREATED_DATE
,CREATED_BY
,SUBLEDGER_FLAG
,subledger_cn
,agency_code
) VALUES (
NULL
,:new.cn
,NULL
,:new.OBJ_NAME
,:new.OBJ_CLASS
,:new.ID
,:new.NAME
,NULL
,NULL --:new.OWNERSHIP
,:new.ADMIN_ORG
,NULL --:new.DEVELOPMENT_STATUS
,:new.CREATED_DATE
,:new.CREATED_BY
,'N'
,null
,:NEW.agency_code -- Added on 10/28/2003 by C.Ma
);

-- UPDATING
ELSIF UPDATING THEN
-- Update table ii_rpa_assets_v_tab
BEGIN
    UPDATE ii_rpa_assets_v_tab tab
    SET (
        OBJECT_NAME
        ,OBJECT_CLASS
        ,ID
        ,NAME
        ,SUB_TYPE
        --,OWNERSHIP
        ,ADMIN_ORG
        ,DEVELOPMENT_STATUS
        ,CREATED_DATE
        ,CREATED_BY
        --,SUBLEDGER_FLAG
    ) = (SELECT
        :new.OBJ_NAME    OBJECT_NAME
        ,:new.OBJ_CLASS  OBJECT_CLASS
        ,:new.ID
        ,:new.NAME
        ,:new.LU_SUBTYPE
        --,:new.OWNERSHIP
        ,:new.ADMIN_ORG
        ,NULL--,:new.DEVELOPMENT_STATUS

```

Oracle Designer Report : cktrigger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

        ,:new.CREATED_DATE
        ,:new.CREATED_BY
        --,'N'
        FROM DUAL)
        WHERE tab.lu_cn = :new.cn;

    EXCEPTION WHEN OTHERS THEN
        ii_rpa_errors.save_log(vModuleName, 'II_RPA_ASSETS_V_TAB',
        :new.cn, null, null, SQLERRM);
    END;

-- Update table ii_rpa_subledgers
-- The corresponding Admin_org in II_RPA_SUBLEDGERS should be
changed too
-- There is a log table via triggers to keep track of all
significant changes
-- occurred to subledger-ed assests.
    BEGIN
        UPDATE ii_rpa_subledgers sub
        SET admin_org = :new.admin_org
        WHERE sub.lu_cn = :new.cn
        AND ii_sql_func.super_equal(admin_org, :new.admin_org) <>
        'TRUE';

    EXCEPTION WHEN OTHERS THEN
        --ii_rpa_errors.log('LAND_UNITS', :new.cn, SQLERRM);
        -- Modified on 01/07/2004
        ii_rpa_errors.save_log(vModuleName, 'LAND_UNITS', :new.cn,
        null, null, SQLERRM);
        RAISE;
    END;

-----
-- Audit
-----

DECLARE
    vOldLu land_units%rowtype;
    vNewLu land_units%rowtype;
BEGIN
    vOldLu.CN := :old.CN;
    vOldLu.OBJ_TECH := :old.OBJ_TECH;
    vOldLu.OBJ_NAME := :old.OBJ_NAME;
    vOldLu.OBJ_CLASS := :old.OBJ_CLASS;
    vOldLu.ID := :old.ID;
    vOldLu.NAME := :old.NAME;
    vOldLu.LU_SUBTYPE := :old.LU_SUBTYPE;
    vOldLu.ADMIN_ORG := :old.ADMIN_ORG;
    vOldLu.MASTER_SITE := :old.MASTER_SITE;
    vOldLu.AVG_ELEVATION := :old.AVG_ELEVATION;
    vOldLu.PREDOMINANT_LANDFORM := :old.PREDOMINANT_LANDFORM;
    vOldLu.STATUS := :old.STATUS;
    vOldLu.AREA_UOM := :old.AREA_UOM;
    vOldLu.TRANS_ID := :old.TRANS_ID;
    vOldLu.AREA := :old.AREA;
    vOldLu.PREDOMNT_SFC_SUBSTRATE := :old.PREDOMNT_SFC_SUBSTRATE;
    vOldLu.AVG_ELEVATION_UOM := :old.AVG_ELEVATION_UOM;
    vOldLu.REMARKS := :old.REMARKS;
    vOldLu.ORG_JUR_CONTACT_CN := :old.ORG_JUR_CONTACT_CN;
    vOldLu.ROOT_CN := :old.ROOT_CN;

```

Oracle Designer Report : cktrigger

Container: Real Property Management Version :

Table/View Name :LAND_UNITS (table)

```

voldLu.PARENT_CN := :old.PARENT_CN;
voldLu.CREATED_BY := :old.CREATED_BY;
voldLu.CREATED_DATE := :old.CREATED_DATE;
voldLu.CREATED_IN_INSTANCE := :old.CREATED_IN_INSTANCE;
voldLu.MODIFIED_BY := :old.MODIFIED_BY;
voldLu.MODIFIED_DATE := :old.MODIFIED_DATE;
voldLu.MODIFIED_IN_INSTANCE := :old.MODIFIED_IN_INSTANCE;
voldLu.OWNERSHIP := :old.OWNERSHIP;
--voldLu.GSA_INSTALLATION_CN := :old.GSA_INSTALLATION_CN;--
commented out on 10/28/2003 by C.Ma

vNewLu.CN := :new.CN;
vNewLu.OBJ_TECH := :new.OBJ_TECH;
vNewLu.OBJ_NAME := :new.OBJ_NAME;
vNewLu.OBJ_CLASS := :new.OBJ_CLASS;
vNewLu.ID := :new.ID;
vNewLu.NAME := :new.NAME;
vNewLu.LU_SUBTYPE := :new.LU_SUBTYPE;
vNewLu.ADMIN_ORG := :new.ADMIN_ORG;
vNewLu.MASTER_SITE := :new.MASTER_SITE;
vNewLu.AVG_ELEVATION := :new.AVG_ELEVATION;
vNewLu.PREDOMINANT_LANDFORM := :new.PREDOMINANT_LANDFORM;
vNewLu.STATUS := :new.STATUS;
vNewLu.AREA_UOM := :new.AREA_UOM;
vNewLu.TRANS_ID := :new.TRANS_ID;
vNewLu.AREA := :new.AREA;
vNewLu.PREDOMNT_SFC_SUBSTRATE := :new.PREDOMNT_SFC_SUBSTRATE;
vNewLu.AVG_ELEVATION_UOM := :new.AVG_ELEVATION_UOM;
vNewLu.REMARKS := :new.REMARKS;
vNewLu.ORG_JUR_CONTACT_CN := :new.ORG_JUR_CONTACT_CN;
vNewLu.ROOT_CN := :new.ROOT_CN;
vNewLu.PARENT_CN := :new.PARENT_CN;
vNewLu.CREATED_BY := :new.CREATED_BY;
vNewLu.CREATED_DATE := :new.CREATED_DATE;
vNewLu.CREATED_IN_INSTANCE := :new.CREATED_IN_INSTANCE;
vNewLu.MODIFIED_BY := :new.MODIFIED_BY;
vNewLu.MODIFIED_DATE := :new.MODIFIED_DATE;
vNewLu.MODIFIED_IN_INSTANCE := :new.MODIFIED_IN_INSTANCE;
vNewLu.OWNERSHIP := :new.OWNERSHIP;
--vNewLu.GSA_INSTALLATION_CN := :new.GSA_INSTALLATION_CN;--
commented out on 10/28/2003 by C.Ma

    ii_rpa_auditing.audit_lu('U', :new.modified_by,
:new.modified_date, voldLu, vNewLu);
    END;

END IF;

END;
```

Container: Real Property Management Version :

Table/View Name :II_CONTACTS (table)

Trigger Name :	CONTACTS_TRG	Create?	Yes
PL/SQL Module			
Name :	CONTACTS_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
		Delete?	No
New Alias	NEW	Update?	Yes
Old Alias	: OLD		

Trigger Logic

```

DECLARE
  vModuleName CONSTANT VARCHAR2(30) := 'II_LU_RPA_AR_TRG';  --
Added on 01/07/2003
BEGIN
  -- DELETING
  IF DELETING THEN
    -- DELETE from ii_rpa_assets_v_tab
    DELETE FROM ii_rpa_assets_v_tab
    WHERE lu_cn = :old.cn;

  -- INSERTING
  ELSIF INSERTING THEN
    -- Insert into table ii_rpa_assets_v_tab
    INSERT INTO ii_rpa_assets_v_tab (
      FEA_CN
    , LU_CN
    , PLD_CN
    , OBJECT_NAME
    , OBJECT_CLASS
    , ID
    , NAME
    , SUB_TYPE
    , OWNERSHIP
    , ADMIN_ORG
    , DEVELOPMENT_STATUS
    , CREATED_DATE
    , CREATED_BY
    , SUBLEDGER_FLAG
    , subledger_cn
    , agency_code
    ) VALUES (
      NULL
    , :new.cn
    , NULL
    , :new.OBJ_NAME
    , :new.OBJ_CLASS
    , :new.ID
    , :new.NAME
    , NULL
    , NULL --:new.OWNERSHIP
    , :new.ADMIN_ORG
    , NULL --:new.DEVELOPMENT_STATUS
  )

```

Oracle Designer Report : ckttrigger

Container: Real Property Management Version :

Table/View Name :II_CONTACTS (table)

```

, :new.CREATED_DATE
, :new.CREATED_BY
, 'N'
, null
, :NEW.agency_code -- Added on 10/28/2003 by C.Ma
);

-- UPDATING
ELSIF UPDATING THEN
-- Update table ii_rpa_assets_v_tab
BEGIN
    UPDATE ii_rpa_assets_v_tab tab
    SET (
        OBJECT_NAME
        , OBJECT_CLASS
        , ID
        , NAME
        , SUB_TYPE
        --, OWNERSHIP
        , ADMIN_ORG
        , DEVELOPMENT_STATUS
        , CREATED_DATE
        , CREATED_BY
        --, SUBLEDGER_FLAG
    ) = (SELECT
        :new.OBJ_NAME   OBJECT_NAME
        , :new.OBJ_CLASS OBJECT_CLASS
        , :new.ID
        , :new.NAME
        , :new.LU SUBTYPE
        --, :new.OWNERSHIP
        , :new.ADMIN_ORG
        , NULL--, :new.DEVELOPMENT_STATUS
        , :new.CREATED_DATE
        , :new.CREATED_BY
        --, 'N'
        FROM DUAL)
    WHERE tab.lu_cn = :new.cn;

    EXCEPTION WHEN OTHERS THEN
        ii_rpa_errors.save_log(vModuleName, 'II_RPA_ASSETS_V_TAB',
        :new.cn, null, null, SQLERRM);
    END;

-- Update table ii_rpa_subledgers
-- The corresponding Admin_org in II_RPA_SUBLEDGERS should be
changed too
-- There is a log table via triggers to keep track of all
significant changes
-- occurred to subledger-ed assests.
BEGIN
    UPDATE ii_rpa_subledgers sub
    SET admin_org = :new.admin_org
    WHERE sub.lu_cn = :new.cn
    AND ii_sql_func.super_equal(admin_org, :new.admin_org) <>
TRUE';

    EXCEPTION WHEN OTHERS THEN
        --ii_rpa_errors.log('LAND_UNITS', :new.cn, SQLERRM);

```

Oracle Designer Report : cktrgger

Container: Real Property Management

Version :

Table/View Name :II_CONTACTS (table)

```

-- Modified on 01/07/2004
ii_rpa_errors.save_log(vModuleName, 'LAND_UNITS', :new.cn,
null, null, SQLERRM);
RAISE;
END;

-----
-- Audit
-----
DECLARE
  voldLu land_units%rowtype;
  vNewLu land_units%rowtype;
BEGIN
  voldLu.CN := :old.CN;
  voldLu.OBJ_TECH := :old.OBJ_TECH;
  voldLu.OBJ_NAME := :old.OBJ_NAME;
  voldLu.OBJ_CLASS := :old.OBJ_CLASS;
  voldLu.ID := :old.ID;
  voldLu.NAME := :old.NAME;
  voldLu.LU_SUBTYPE := :old.LU_SUBTYPE;
  voldLu.ADMIN_ORG := :old.ADMIN_ORG;
  voldLu.MASTER_SITE := :old.MASTER_SITE;
  voldLu.AVG_ELEVATION := :old.AVG_ELEVATION;
  voldLu.PREDOMINANT_LANDFORM := :old.PREDOMINANT_LANDFORM;
  voldLu.STATUS := :old.STATUS;
  voldLu.AREA_UOM := :old.AREA_UOM;
  voldLu.TRANS_ID := :old.TRANS_ID;
  voldLu.AREA := :old.AREA;
  voldLu.PREDOMNT_SFC_SUBSTRATE := :old.PREDOMNT_SFC_SUBSTRATE;
  voldLu.AVG_ELEVATION_UOM := :old.AVG_ELEVATION_UOM;
  voldLu.REMARKS := :old.REMARKS;
  voldLu.ORG_JUR_CONTACT_CN := :old.ORG_JUR_CONTACT_CN;
  voldLu.ROOT_CN := :old.ROOT_CN;
  voldLu.PARENT_CN := :old.PARENT_CN;
  voldLu.CREATED_BY := :old.CREATED_BY;
  voldLu.CREATED_DATE := :old.CREATED_DATE;
  voldLu.CREATED_IN_INSTANCE := :old.CREATED_IN_INSTANCE;
  voldLu.MODIFIED_BY := :old.MODIFIED_BY;
  voldLu.MODIFIED_DATE := :old.MODIFIED_DATE;
  voldLu.MODIFIED_IN_INSTANCE := :old.MODIFIED_IN_INSTANCE;
  voldLu.OWNERSHIP := :old.OWNERSHIP;
  --voldLu.GSA_INSTALLATION_CN := :old.GSA_INSTALLATION_CN;--
  commented out on 10/28/2003 by C.Ma

  vNewLu.CN := :new.CN;
  vNewLu.OBJ_TECH := :new.OBJ_TECH;
  vNewLu.OBJ_NAME := :new.OBJ_NAME;
  vNewLu.OBJ_CLASS := :new.OBJ_CLASS;
  vNewLu.ID := :new.ID;
  vNewLu.NAME := :new.NAME;
  vNewLu.LU_SUBTYPE := :new.LU_SUBTYPE;
  vNewLu.ADMIN_ORG := :new.ADMIN_ORG;
  vNewLu.MASTER_SITE := :new.MASTER_SITE;
  vNewLu.AVG_ELEVATION := :new.AVG_ELEVATION;
  vNewLu.PREDOMINANT_LANDFORM := :new.PREDOMINANT_LANDFORM;
  vNewLu.STATUS := :new.STATUS;
  vNewLu.AREA_UOM := :new.AREA_UOM;
  vNewLu.TRANS_ID := :new.TRANS_ID;
  vNewLu.AREA := :new.AREA;

```

Oracle Designer Report : cktrgger

Container: Real Property Management

Version :

Table/View Name :II_CONTACTS (table)

```

vNewLu.PREDOMNT_SFC_SUBSTRATE := :new.PREDOMNT_SFC_SUBSTRATE;
vNewLu.AVG_ELEVATION_UOM := :new.AVG_ELEVATION_UOM;
vNewLu.REMARKS := :new.REMARKS;
vNewLu.ORG_JUR_CONTACT_CN := :new.ORG_JUR_CONTACT_CN;
vNewLu.ROOT_CN := :new.ROOT_CN;
vNewLu.PARENT_CN := :new.PARENT_CN;
vNewLu.CREATED_BY := :new.CREATED_BY;
vNewLu.CREATED_DATE := :new.CREATED_DATE;
vNewLu.CREATED_IN_INSTANCE := :new.CREATED_IN_INSTANCE;
vNewLu.MODIFIED_BY := :new.MODIFIED_BY;
vNewLu.MODIFIED_DATE := :new.MODIFIED_DATE;
vNewLu.MODIFIED_IN_INSTANCE := :new.MODIFIED_IN_INSTANCE;
vNewLu.OWNERSHIP := :new.OWNERSHIP;
--vNewLu.GSA_INSTALLATION_CN := :new.GSA_INSTALLATION_CN;--
commented out on 10/28/2003 by C.Ma

    ii_rpa_auditing.audit_lu('U', :new.modified_by,
:new.modified_date, vOldLu, vNewLu);
    END;

END IF;

END;

DECLARE
BEGIN
    DECLARE
        vContCN ii_contacts.cn%TYPE := null;
        vAgency ii_contacts.agency_code%TYPE := null;

        cursor cFSOrg is
            select 'x'
            from ii_organizations
            where org_type = 'FS ORGANIZATION' and
                  cont_cn = vContCN;
        vDummy char(1);

        cursor user_agency is
            select agency_code
            from ii_rpa_fiscal_users
            where user_name = USER;
    BEGIN
        IF INSERTING THEN
            IF :new.cn IS NULL THEN
                :new.cn := db_instance.generate_global_id('CONT_CN_SEQ');
            END IF;
            db_instance.insert_audit_columns
                (:new.created_by, :new.created_in_instance,
:new.created_date);

            IF :new.agency_code IS NULL THEN
                open user_agency;
                fetch user_agency into vAgency;
                close user_agency;
                :new.agency_code := vAgency;
            END IF;

            ELSE -- Updating

```

Oracle Designer Report : cktggr

Container: Real Property Management

Version :

Table/View Name :II_CONTACTS (table)

```

-- Check role assignment for FS Org updates.
if not ii_chk_role_assign('INFRA_SVC_MGR') then
-- Make sure user isn't trying to update an FS Org.
vContCN := :old.cn;
open cFSOrg;
fetch cFSOrg into vDummy;
if not cFSOrg%notfound then
close cFSOrg;
raise_application_error(-20300,'Insufficient privileges
to '||
        'update an FS Organization');
else
close cFSOrg;
end if;
end if;
--
-- Set audit columns
db_instance.update_audit_columns
(:new.modified_By, :new.modified_in_instance,
:new.modified_date);
--
-- Added by RAdams on 6/24/2002 for v5.0:
--
-- Record old/new ID and CN in PL/SQL table for purposes of
-- writing row to range permits audit table
(ii_range_permits_audit) when the updated
-- Contact ID is a permittee or managing organization for a
-- grazing permit. This action takes place in the statement-
level
-- after-update/delete trigger that first queries the range
permits view
-- to check if this contact is a permittee or managing
organization for
-- a grazing permit. If this is the case, a re-query on the
ii_contacts table is necessary
-- to get the remaining Permittee or Managing Org ID to be
inserted
-- into the audit table. To avoid the "mutating table" error,
the re-query of the ii_contacts
-- table needed to be moved to a statement-level after
update/delete trigger, passing "old" and "new"
-- values in the package PL/SQL table referenced in the code
below.
--
if nvl(:new.id,'@') != nvl(:old.id,'@') then
ii_range_permits_audit.vNumEntries :=
ii_range_permits_audit.vNumEntries + 1;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ne
w_id := :new.id;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ol
d_id := :old.id;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ne
w_cn := :new.cn;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ol
d_cn := :old.cn;

```

Oracle Designer Report : cktrgger

Container: Real Property Management **Version :**

Table/View Name :II_CONTACTS (table)

```
        end if;
        --
        END IF;
    END;
END;
```

Container: Real Property Management Version :

Table/View Name :II_FEATURES (table)

Trigger Name : II_RBIU_FEATURES Create? Y

PL/SQL Module
Name : II_RBIU_FEATURES_1 Enabled? Y

PL/SQL File :

Purpose :

Trigger Time: BEFORE Level: ROW Insert? Y

Delete? N

New Alias Update? Y

Old Alias :

Trigger Logic

```

DECLARE
BEGIN
  DECLARE
    vContCN ii_contacts.cn%TYPE := null;
    vAgency ii_contacts.agency_code%TYPE := null;

    cursor cFSOrg is
      select 'x'
      from ii_organizations
      where org_type = 'FS ORGANIZATION' and
            cont_cn = vContCN;
    vDummy char(1);

    cursor user_agency is
      select agency_code
      from ii_rpa_fiscal_users
      where user_name = USER;
  BEGIN
    IF INSERTING THEN
      IF :new.cn IS NULL THEN
        :new.cn := db_instance.generate_global_id('CONT_CN_SEQ');
      END IF;
      db_instance.insert_audit_columns
        (:new.created_by, :new.created_in_instance,
        :new.created_date);

      IF :new.agency_code IS NULL THEN
        open user_agency;
        fetch user_agency into vAgency;
        close user_agency;
        :new.agency_code := vAgency;
      END IF;

    ELSE -- Updating
      -- Check role assignment for FS Org updates.
      if not ii_chk_role_assign('INFRA_SVC_MGR') then
        -- Make sure user isn't trying to update an FS Org.
        vContCN := :old.cn;
        open cFSOrg;
        fetch cFSOrg into vDummy;
        if not cFSOrg%notfound then
          close cFSOrg;

```

Oracle Designer Report : cktrigger

Container: Real Property Management Version :

Table/View Name :II_FEATURES (table)

```

        raise_application_error(-20300,'Insufficient privileges
to '||
        'update an FS Organization');
    else
        close cFSOrg;
    end if;
end if;
--
-- Set audit columns
db_instance.update_audit_columns
(:new.modified_by, :new.modified_in_instance,
:new.modified_date);
--
-- Added by RAdams on 6/24/2002 for v5.0:
--
-- Record old/new ID and CN in PL/SQL table for purposes of
-- writing row to range permits audit table
(ii_range_permit_audits) when the updated
-- Contact ID is a permittee or managing organization for a
-- grazing permit. This action takes place in the statement-
level
-- after-update/delete trigger that first queries the range
permits view
-- to check if this contact is a permittee or managing
organization for
-- a grazing permit. If this is the case, a re-query on the
ii_contacts table is necessary
-- to get the remaining Permittee or Managing Org ID to be
inserted
-- into the audit table. To avoid the "mutating table" error,
the re-query of the ii_contacts
-- table needed to be moved to a statement-level after
update/delete trigger, passing "old" and "new"
-- values in the package PL/SQL table referenced in the code
below.
--
    if nvl(:new.id,'@') != nvl(:old.id,'@') then
        ii_range_permits_audit.vNumEntries :=
ii_range_permits_audit.vNumEntries + 1;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ne
w_id := :new.id;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ol
d_id := :old.id;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ne
w_cn := :new.cn;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ol
d_cn := :old.cn;
    end if;
--
    END IF;
END;
END;

DECLARE
```

Oracle Designer Report : cktgrgger

Container: Real Property Management Version :

Table/View Name :II_FEATURES (table)

```

BEGIN
  DECLARE
    vContCN ii_contacts.cn%TYPE := null;
    vAgency ii_contacts.agency_code%TYPE := null;

    cursor cFSOrg is
      select 'x'
      from ii_organizations
      where org_type = 'FS ORGANIZATION' and
            cont_cn = vContCN;
    vDummy char(1);

    cursor user_agency is
      select agency_code
      from ii_rpa_fiscal_users
      where user_name = USER;
  BEGIN
    IF INSERTING THEN
      IF :new.cn IS NULL THEN
        :new.cn := db_instance.generate_global_id('CONT_CN_SEQ');
      END IF;
      db_instance.insert_audit_columns
        (:new.created_by, :new.created_in_instance,
        :new.created_date);

      IF :new.agency_code IS NULL THEN
        open user_agency;
        fetch user_agency into vAgency;
        close user_agency;
        :new.agency_code := vAgency;
      END IF;

    ELSE -- Updating
      -- Check role assignment for FS Org updates.
      if not ii_chk_role_assign('INFRA_SVC_MGR') then
        -- Make sure user isn't trying to update an FS Org.
        vContCN := :old.cn;
        open cFSOrg;
        fetch cFSOrg into vDummy;
        if not cFSOrg%notfound then
          close cFSOrg;
          raise_application_error(-20300,'Insufficient privileges
to '||
              'update an FS Organization');
        else
          close cFSOrg;
        end if;
      end if;
      --
      -- Set audit columns
      db_instance.update_audit_columns
        (:new.modified_by, :new.modified_in_instance,
        :new.modified_date);
      --
      -- Added by RAdams on 6/24/2002 for v5.0:
      --
      -- Record old/new ID and CN in PL/SQL table for purposes of
      -- writing row to range permits audit table
      (ii_range_permit_audits) when the updated

```

Oracle Designer Report : cktrgger

Container: Real Property Management

Version :

Table/View Name :II_FEATURES (table)

```

-- Contact ID is a permittee or managing organization for a
-- grazing permit. This action takes place in the statement-
level
-- after-update/delete trigger that first queries the range
permits view
-- to check if this contact is a permittee or managing
organization for
-- a grazing permit. If this is the case, a re-query on the
ii_contacts table is necessary
-- to get the remaining Permittee or Managing Org ID to be
inserted
-- into the audit table. To avoid the "mutating table" error,
the re-query of the ii_contacts
-- table needed to be moved to a statement-level after
update/delete trigger, passing "old" and "new"
-- values in the package PL/SQL table referenced in the code
below.
--
if nvl(:new.id, '@') != nvl(:old.id, '@') then
    ii_range_permits_audit.vNumEntries :=
ii_range_permits_audit.vNumEntries + 1;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ne
w_id := :new.id;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ol
d_id := :old.id;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ne
w_cn := :new.cn;

ii_range_permits_audit.vPRAudit(ii_range_permits_audit.vNumEntries).ol
d_cn := :old.cn;
end if;
--
END IF;
END;
END;

DECLARE
BEGIN
    IF :new.OWNERSHIP is not null and
    :new.OWNERSHIP <> nvl(:old.OWNERSHIP, '@') and
    NOT Chk_IiCode( 'FEAT_OWNERSHIP', :new.OWNERSHIP )
    THEN
        Raise_Application_Error( -20301,
            'Invalid value "'||:new.OWNERSHIP||'" '||
            'for II_Features.OWNERSHIP.' );
    END IF;
    --
    IF :new.MAINTENANCE_LEVEL is not null and
    :new.MAINTENANCE_LEVEL <> nvl(:old.MAINTENANCE_LEVEL, '@') and
    NOT Chk_IiCode( 'FEAT_MAINTENANCE_LEVEL', :new.MAINTENANCE_LEVEL
    )
    THEN
        Raise_Application_Error( -20302,
            'Invalid value "'||:new.MAINTENANCE_LEVEL||'" '||
            'for II_Features.MAINTENANCE_LEVEL.' );
    END IF;

```

Oracle Designer Report : cktgrger

Container: Real Property Management Version :

Table/View Name :II_FEATURES (table)

```

END IF;
--
IF :new.MAINTENANCE_RESPONSIBILITY is not null and
:new.MAINTENANCE_RESPONSIBILITY <>
nvl(:old.MAINTENANCE_RESPONSIBILITY, '@') and
NOT Chk_IiCode( 'MAINTENANCE_RESPONSIBILITY',
:new.MAINTENANCE_RESPONSIBILITY )
THEN
  Raise_Application_Error( -20303,
    'Invalid value "'||:new.MAINTENANCE_RESPONSIBILITY||'" '||
    'for II_Features.MAINTENANCE_RESPONSIBILITY.' );
END IF;
--
IF :new.CONDITION_RATING is not null and
:new.CONDITION_RATING <> nvl(:old.CONDITION_RATING, '@') and
NOT Chk_IiCode( 'FEAT_CONDITION_RATING', :new.CONDITION_RATING )
THEN
  Raise_Application_Error( -20304,
    'Invalid value "'||:new.CONDITION_RATING||'" '||
    'for II_Features.CONDITION_RATING.' );
END IF;
--
IF :new.APPRAISAL_RATING is not null and
:new.APPRAISAL_RATING <> nvl(:old.APPRAISAL_RATING, '@') and
NOT Chk_IiCode( 'FEAT_APPRAISAL_RATING', :new.APPRAISAL_RATING )
THEN
  Raise_Application_Error( -20305,
    'Invalid value "'||:new.APPRAISAL_RATING||'" '||
    'for II_Features.APPRAISAL_RATING.' );
END IF;
--
IF :new.DEVELOPMENT_STATUS is not null and
:new.DEVELOPMENT_STATUS <> nvl(:old.DEVELOPMENT_STATUS, '@') and
NOT Chk_IiCode( 'FEAT_DEVELOPMENT_STATUS',
:new.DEVELOPMENT_STATUS )
THEN
  Raise_Application_Error( -20306,
    'Invalid value "'||:new.DEVELOPMENT_STATUS||'" '||
    'for II_Features.DEVELOPMENT_STATUS.' );
END IF;
--
IF :new.HISTORICAL_STATUS is not null and
:new.HISTORICAL_STATUS <> nvl(:old.HISTORICAL_STATUS, '@') and
NOT Chk_IiCode( 'FEAT_HISTORICAL_STATUS', :new.HISTORICAL_STATUS
)
THEN
  Raise_Application_Error( -20307,
    'Invalid value "'||:new.HISTORICAL_STATUS||'" '||
    'for II_Features.HISTORICAL_STATUS.' );
END IF;
--
IF :new.LOCATION is not null and
:new.LOCATION <> nvl(:old.LOCATION, '@') and
NOT Chk_IiCode( 'FEAT_LOCATION', :new.LOCATION )
THEN
  Raise_Application_Error( -20308,
    'Invalid value "'||:new.LOCATION||'" '||
    'for II_Features.LOCATION.' );
END IF;

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :II_FEATURES (table)

```

--
IF :new.LOCATION_UOM is not null and
:new.LOCATION_UOM <> nvl(:old.LOCATION_UOM,'@') and
NOT Chk_IiCode( 'FEAT_UOM', :new.LOCATION_UOM )
THEN
  Raise_Application_Error( -20309,
    'Invalid value "'||:new.LOCATION_UOM||'" '||
    'for II_Features.LOCATION_UOM.' );
END IF;
--
IF :new.ACCESSIBILITY_STATUS is not null and
:new.ACCESSIBILITY_STATUS <>
nvl(:old.ACCESSIBILITY_STATUS,'@') and
NOT Chk_IiCode( 'ACCESSIBILITY_STATUS', :new.ACCESSIBILITY_STATUS
)
THEN
  Raise_Application_Error( -203010,
    'Invalid value "'||:new.ACCESSIBILITY_STATUS||'" '||
    'for II_Features.ACCESSIBILITY_STATUS.' );
END IF;
--
IF :new.ACQUISITION_METHOD is not null and
:new.ACQUISITION_METHOD <> nvl(:old.ACQUISITION_METHOD,'@') and
NOT Chk_IiCode( 'ACQUISITION_METHOD', :new.ACQUISITION_METHOD )
THEN
  Raise_Application_Error( -203011,
    'Invalid value "'||:new.ACQUISITION_METHOD||'" '||
    'for II_Features.ACQUISITION_METHOD.' );
END IF;
--
IF :new.OPERATIONAL_STATUS is not null and
:new.OPERATIONAL_STATUS <> nvl(:old.OPERATIONAL_STATUS,'@') and
NOT Chk_IiCode( 'OPERATIONAL_STATUS', :new.OPERATIONAL_STATUS )
THEN
  Raise_Application_Error( -203012,
    'Invalid value "'||:new.OPERATIONAL_STATUS||'" '||
    'for II_Features.OPERATIONAL_STATUS.' );
END IF;
--
IF Inserting
THEN
  -- Supply CN if it is null.
  IF :new.CN is null
  THEN
    :new.CN := Db_Instance.GENERATE_GLOBAL_ID( 'FEAT_CN_SEQ' );
  END IF;
  --
  -- Set the CREATED... audit columns on row insert if not
  supplied.
  IF :new.CREATED_BY is null or
  :new.CREATED_DATE is null or
  :new.CREATED_IN_INSTANCE is null
  THEN
    Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
                                     , :new.CREATED_IN_INSTANCE
                                     , :new.CREATED_DATE );
  END IF;
  ---
ELSIF Updating

```

Oracle Designer Report : cktrigger

Container: Real Property Management **Version :**

Table/View Name :II_FEATURES (table)

```
THEN
  -- Set the MODIFIED... audit columns on row update.
  Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY
                                   , :new.MODIFIED_IN_INSTANCE
                                   , :new.MODIFIED_DATE );
  ---
END IF;
END;
```

Container: Real Property Management

Version :

Table/View Name :II_FEATURES (table)

Trigger Name :	II_FEA_RPA_AR_TRG	Create? Yes
PL/SQL Module Name :	II_FEA_RPA_AR_TRG_1	Enabled? Yes
PL/SQL File :		
Purpose :	Maintain Ins/Upd/Del after row for Real Property Accounting	
Trigger Time:	AFTER	Level: ROW
		Insert? Yes
		Delete? Yes
New Alias		Update? Yes
Old Alias :		

Trigger Logic

```

DECLARE
BEGIN
  IF :new.OWNERSHIP is not null and
    :new.OWNERSHIP <> nvl(:old.OWNERSHIP,'@') and
    NOT Chk_IiCode( 'FEAT_OWNERSHIP', :new.OWNERSHIP )
  THEN
    Raise_Application_Error( -20301,
      'Invalid value "'||:new.OWNERSHIP||'" '||
      'for II_Features.OWNERSHIP.' );
  END IF;
  --
  IF :new.MAINTENANCE_LEVEL is not null and
    :new.MAINTENANCE_LEVEL <> nvl(:old.MAINTENANCE_LEVEL,'@') and
    NOT Chk_IiCode( 'FEAT_MAINTENANCE_LEVEL', :new.MAINTENANCE_LEVEL
  )
  THEN
    Raise_Application_Error( -20302,
      'Invalid value "'||:new.MAINTENANCE_LEVEL||'" '||
      'for II_Features.MAINTENANCE_LEVEL.' );
  END IF;
  --
  IF :new.MAINTENANCE_RESPONSIBILITY is not null and
    :new.MAINTENANCE_RESPONSIBILITY <>
    nvl(:old.MAINTENANCE_RESPONSIBILITY,'@') and
    NOT Chk_IiCode( 'MAINTENANCE_RESPONSIBILITY',
      :new.MAINTENANCE_RESPONSIBILITY )
  THEN
    Raise_Application_Error( -20303,
      'Invalid value "'||:new.MAINTENANCE_RESPONSIBILITY||'" '||
      'for II_Features.MAINTENANCE_RESPONSIBILITY.' );
  END IF;
  --
  IF :new.CONDITION_RATING is not null and
    :new.CONDITION_RATING <> nvl(:old.CONDITION_RATING,'@') and
    NOT Chk_IiCode( 'FEAT_CONDITION_RATING', :new.CONDITION_RATING )
  THEN
    Raise_Application_Error( -20304,
      'Invalid value "'||:new.CONDITION_RATING||'" '||
      'for II_Features.CONDITION_RATING.' );
  END IF;
  --

```

Oracle Designer Report : ckttrigger

Container: Real Property Management Version :

Table/View Name :II_FEATURES (table)

```

IF :new.APPRAISAL_RATING is not null and
:new.APPRAISAL_RATING <> nvl(:old.APPRAISAL_RATING,'@') and
NOT Chk_IiCode( 'FEAT_APPRAISAL_RATING', :new.APPRAISAL_RATING )
THEN
  Raise_Application_Error( -20305,
    'Invalid value "'||:new.APPRAISAL_RATING||'" '||
    'for II_Features.APPRAISAL_RATING.' );
END IF;
--
IF :new.DEVELOPMENT_STATUS is not null and
:new.DEVELOPMENT_STATUS <> nvl(:old.DEVELOPMENT_STATUS,'@') and
NOT Chk_IiCode( 'FEAT_DEVELOPMENT_STATUS',
:new.DEVELOPMENT_STATUS )
THEN
  Raise_Application_Error( -20306,
    'Invalid value "'||:new.DEVELOPMENT_STATUS||'" '||
    'for II_Features.DEVELOPMENT_STATUS.' );
END IF;
--
IF :new.HISTORICAL_STATUS is not null and
:new.HISTORICAL_STATUS <> nvl(:old.HISTORICAL_STATUS,'@') and
NOT Chk_IiCode( 'FEAT_HISTORICAL_STATUS', :new.HISTORICAL_STATUS
)
THEN
  Raise_Application_Error( -20307,
    'Invalid value "'||:new.HISTORICAL_STATUS||'" '||
    'for II_Features.HISTORICAL_STATUS.' );
END IF;
--
IF :new.LOCATION is not null and
:new.LOCATION <> nvl(:old.LOCATION,'@') and
NOT Chk_IiCode( 'FEAT_LOCATION', :new.LOCATION )
THEN
  Raise_Application_Error( -20308,
    'Invalid value "'||:new.LOCATION||'" '||
    'for II_Features.LOCATION.' );
END IF;
--
IF :new.LOCATION_UOM is not null and
:new.LOCATION_UOM <> nvl(:old.LOCATION_UOM,'@') and
NOT Chk_IiCode( 'FEAT_UOM', :new.LOCATION_UOM )
THEN
  Raise_Application_Error( -20309,
    'Invalid value "'||:new.LOCATION_UOM||'" '||
    'for II_Features.LOCATION_UOM.' );
END IF;
--
IF :new.ACCESSIBILITY_STATUS is not null and
:new.ACCESSIBILITY_STATUS <>
nvl(:old.ACCESSIBILITY_STATUS,'@') and
NOT Chk_IiCode( 'ACCESSIBILITY_STATUS', :new.ACCESSIBILITY_STATUS
)
THEN
  Raise_Application_Error( -203010,
    'Invalid value "'||:new.ACCESSIBILITY_STATUS||'" '||
    'for II_Features.ACCESSIBILITY_STATUS.' );
END IF;
--
IF :new.ACQUISITION_METHOD is not null and

```

Oracle Designer Report : cktrigger

Container: Real Property Management

Version :

Table/View Name :II_FEATURES (table)

```

: new.ACQUISITION_METHOD <> nvl(:old.ACQUISITION_METHOD, '@') and
NOT Chk_IiCode( 'ACQUISITION_METHOD', :new.ACQUISITION_METHOD )
THEN
  Raise_Application_Error( -203011,
    'Invalid value "'||:new.ACQUISITION_METHOD||'" '||
    'for II_Features.ACQUISITION_METHOD.' );
END IF;
--
IF :new.OPERATIONAL_STATUS is not null and
: new.OPERATIONAL_STATUS <> nvl(:old.OPERATIONAL_STATUS, '@') and
NOT Chk_IiCode( 'OPERATIONAL_STATUS', :new.OPERATIONAL_STATUS )
THEN
  Raise_Application_Error( -203012,
    'Invalid value "'||:new.OPERATIONAL_STATUS||'" '||
    'for II_Features.OPERATIONAL_STATUS.' );
END IF;
--
IF Inserting
THEN
  -- Supply CN if it is null.
  IF :new.CN is null
  THEN
    :new.CN := Db_Instance.GENERATE_GLOBAL_ID( 'FEAT_CN_SEQ' );
  END IF;
  --
  -- Set the CREATED... audit columns on row insert if not
  supplied.
  IF :new.CREATED_BY is null or
  :new.CREATED_DATE is null or
  :new.CREATED_IN_INSTANCE is null
  THEN
    Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
                                     , :new.CREATED_IN_INSTANCE
                                     , :new.CREATED_DATE );
  END IF;
  ---
ELSIF Updating
THEN
  -- Set the MODIFIED... audit columns on row update.
  Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY
                                   , :new.MODIFIED_IN_INSTANCE
                                   , :new.MODIFIED_DATE );
  ---
END IF;
END;
DECLARE
BEGIN
  IF :new.OWNERSHIP is not null and
  :new.OWNERSHIP <> nvl(:old.OWNERSHIP, '@') and
  NOT Chk_IiCode( 'FEAT_OWNERSHIP', :new.OWNERSHIP )
  THEN
    Raise_Application_Error( -20301,
      'Invalid value "'||:new.OWNERSHIP||'" '||
      'for II_Features.OWNERSHIP.' );
  END IF;
  --
  IF :new.MAINTENANCE_LEVEL is not null and
  :new.MAINTENANCE_LEVEL <> nvl(:old.MAINTENANCE_LEVEL, '@') and

```

Oracle Designer Report : cktggr

Container: Real Property Management

Version :

Table/View Name :II_FEATURES (table)

```

        NOT Chk_IiCode( 'FEAT_MAINTENANCE_LEVEL', :new.MAINTENANCE_LEVEL
    )
    THEN
        Raise_Application_Error( -20302,
            'Invalid value "'||:new.MAINTENANCE_LEVEL||'" '||
            'for II_Features.MAINTENANCE_LEVEL.' );
    END IF;
    --
    IF :new.MAINTENANCE_RESPONSIBILITY is not null and
        :new.MAINTENANCE_RESPONSIBILITY <>
        nvl(:old.MAINTENANCE_RESPONSIBILITY,'@') and
        NOT Chk_IiCode( 'MAINTENANCE_RESPONSIBILITY',
            :new.MAINTENANCE_RESPONSIBILITY )
    THEN
        Raise_Application_Error( -20303,
            'Invalid value "'||:new.MAINTENANCE_RESPONSIBILITY||'" '||
            'for II_Features.MAINTENANCE_RESPONSIBILITY.' );
    END IF;
    --
    IF :new.CONDITION_RATING is not null and
        :new.CONDITION_RATING <> nvl(:old.CONDITION_RATING,'@') and
        NOT Chk_IiCode( 'FEAT_CONDITION_RATING', :new.CONDITION_RATING )
    THEN
        Raise_Application_Error( -20304,
            'Invalid value "'||:new.CONDITION_RATING||'" '||
            'for II_Features.CONDITION_RATING.' );
    END IF;
    --
    IF :new.APPRAISAL_RATING is not null and
        :new.APPRAISAL_RATING <> nvl(:old.APPRAISAL_RATING,'@') and
        NOT Chk_IiCode( 'FEAT_APPRAISAL_RATING', :new.APPRAISAL_RATING )
    THEN
        Raise_Application_Error( -20305,
            'Invalid value "'||:new.APPRAISAL_RATING||'" '||
            'for II_Features.APPRAISAL_RATING.' );
    END IF;
    --
    IF :new.DEVELOPMENT_STATUS is not null and
        :new.DEVELOPMENT_STATUS <> nvl(:old.DEVELOPMENT_STATUS,'@') and
        NOT Chk_IiCode( 'FEAT_DEVELOPMENT_STATUS',
: new.DEVELOPMENT_STATUS )
    THEN
        Raise_Application_Error( -20306,
            'Invalid value "'||:new.DEVELOPMENT_STATUS||'" '||
            'for II_Features.DEVELOPMENT_STATUS.' );
    END IF;
    --
    IF :new.HISTORICAL_STATUS is not null and
        :new.HISTORICAL_STATUS <> nvl(:old.HISTORICAL_STATUS,'@') and
        NOT Chk_IiCode( 'FEAT_HISTORICAL_STATUS', :new.HISTORICAL_STATUS
    )
    THEN
        Raise_Application_Error( -20307,
            'Invalid value "'||:new.HISTORICAL_STATUS||'" '||
            'for II_Features.HISTORICAL_STATUS.' );
    END IF;
    --
    IF :new.LOCATION is not null and
        :new.LOCATION <> nvl(:old.LOCATION,'@') and

```

Oracle Designer Report : ckttrigger

Container: Real Property Management

Version :

Table/View Name :II_FEATURES (table)

```

NOT Chk_IiCode( 'FEAT_LOCATION', :new.LOCATION )
THEN
  Raise_Application_Error( -20308,
    'Invalid value "'||:new.LOCATION||'" '||
    'for II_Features.LOCATION.' );
END IF;
--
IF :new.LOCATION_UOM is not null and
:new.LOCATION_UOM <> nvl(:old.LOCATION_UOM, '@') and
NOT Chk_IiCode( 'FEAT_UOM', :new.LOCATION_UOM )
THEN
  Raise_Application_Error( -20309,
    'Invalid value "'||:new.LOCATION_UOM||'" '||
    'for II_Features.LOCATION_UOM.' );
END IF;
--
IF :new.ACCESSIBILITY_STATUS is not null and
:new.ACCESSIBILITY_STATUS <>
nvl(:old.ACCESSIBILITY_STATUS, '@') and
NOT Chk_IiCode( 'ACCESSIBILITY_STATUS', :new.ACCESSIBILITY_STATUS
)
THEN
  Raise_Application_Error( -203010,
    'Invalid value "'||:new.ACCESSIBILITY_STATUS||'" '||
    'for II_Features.ACCESSIBILITY_STATUS.' );
END IF;
--
IF :new.ACQUISITION_METHOD is not null and
:new.ACQUISITION_METHOD <> nvl(:old.ACQUISITION_METHOD, '@') and
NOT Chk_IiCode( 'ACQUISITION_METHOD', :new.ACQUISITION_METHOD )
THEN
  Raise_Application_Error( -203011,
    'Invalid value "'||:new.ACQUISITION_METHOD||'" '||
    'for II_Features.ACQUISITION_METHOD.' );
END IF;
--
IF :new.OPERATIONAL_STATUS is not null and
:new.OPERATIONAL_STATUS <> nvl(:old.OPERATIONAL_STATUS, '@') and
NOT Chk_IiCode( 'OPERATIONAL_STATUS', :new.OPERATIONAL_STATUS )
THEN
  Raise_Application_Error( -203012,
    'Invalid value "'||:new.OPERATIONAL_STATUS||'" '||
    'for II_Features.OPERATIONAL_STATUS.' );
END IF;
--
IF Inserting
THEN
  -- Supply CN if it is null.
  IF :new.CN is null
  THEN
    :new.CN := Db_Instance.GENERATE_GLOBAL_ID( 'FEAT_CN_SEQ' );
  END IF;
  --
  -- Set the CREATED... audit columns on row insert if not
  supplied.
  IF :new.CREATED_BY is null or
    :new.CREATED_DATE is null or
    :new.CREATED_IN_INSTANCE is null
  THEN

```

Oracle Designer Report : cktrigger

Container: Real Property Management Version :

Table/View Name :II_FEATURES (table)

```

        Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
                                           , :new.CREATED_IN INSTANCE
                                           , :new.CREATED_DATE );
    END IF;
    ---
    ELSIF Updating
    THEN
        -- Set the MODIFIED... audit columns on row update.
        Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED BY
                                           , :new.MODIFIED_IN INSTANCE
                                           , :new.MODIFIED_DATE );
        ---
    END IF;
END;
DECLARE
BEGIN
    IF :new.OWNERSHIP is not null and
       :new.OWNERSHIP <> nvl(:old.OWNERSHIP,'@') and
       NOT Chk_IiCode( 'FEAT_OWNERSHIP', :new.OWNERSHIP )
    THEN
        Raise_Application_Error( -20301,
            'Invalid value "'||:new.OWNERSHIP||'" '||
            'for II_Features.OWNERSHIP.' );
    END IF;
    --
    IF :new.MAINTENANCE_LEVEL is not null and
       :new.MAINTENANCE_LEVEL <> nvl(:old.MAINTENANCE_LEVEL,'@') and
       NOT Chk_IiCode( 'FEAT_MAINTENANCE_LEVEL', :new.MAINTENANCE_LEVEL
    )
    THEN
        Raise_Application_Error( -20302,
            'Invalid value "'||:new.MAINTENANCE_LEVEL||'" '||
            'for II_Features.MAINTENANCE_LEVEL.' );
    END IF;
    --
    IF :new.MAINTENANCE_RESPONSIBILITY is not null and
       :new.MAINTENANCE_RESPONSIBILITY <>
       nvl(:old.MAINTENANCE_RESPONSIBILITY,'@') and
       NOT Chk_IiCode( 'MAINTENANCE_RESPONSIBILITY',
           :new.MAINTENANCE_RESPONSIBILITY )
    THEN
        Raise_Application_Error( -20303,
            'Invalid value "'||:new.MAINTENANCE_RESPONSIBILITY||'" '||
            'for II_Features.MAINTENANCE_RESPONSIBILITY.' );
    END IF;
    --
    IF :new.CONDITION_RATING is not null and
       :new.CONDITION_RATING <> nvl(:old.CONDITION_RATING,'@') and
       NOT Chk_IiCode( 'FEAT_CONDITION_RATING', :new.CONDITION_RATING )
    THEN
        Raise_Application_Error( -20304,
            'Invalid value "'||:new.CONDITION_RATING||'" '||
            'for II_Features.CONDITION_RATING.' );
    END IF;
    --
    IF :new.APPRAISAL_RATING is not null and
       :new.APPRAISAL_RATING <> nvl(:old.APPRAISAL_RATING,'@') and
       NOT Chk_IiCode( 'FEAT_APPRAISAL_RATING', :new.APPRAISAL_RATING )

```

Oracle Designer Report : ckttrigger

Container: Real Property Management Version :

Table/View Name :II_FEATURES (table)

```

THEN
    Raise_Application_Error( -20305,
        'Invalid value "'||:new.APPRAISAL_RATING||'" '||
        'for II_Features.APPRAISAL_RATING.' );
END IF;
--
IF :new.DEVELOPMENT_STATUS is not null and
:new.DEVELOPMENT_STATUS <> nvl(:old.DEVELOPMENT_STATUS, '@') and
NOT Chk_IiCode( 'FEAT_DEVELOPMENT_STATUS',
:new.DEVELOPMENT_STATUS )
THEN
    Raise_Application_Error( -20306,
        'Invalid value "'||:new.DEVELOPMENT_STATUS||'" '||
        'for II_Features.DEVELOPMENT_STATUS.' );
END IF;
--
IF :new.HISTORICAL_STATUS is not null and
:new.HISTORICAL_STATUS <> nvl(:old.HISTORICAL_STATUS, '@') and
NOT Chk_IiCode( 'FEAT_HISTORICAL_STATUS', :new.HISTORICAL_STATUS
)
THEN
    Raise_Application_Error( -20307,
        'Invalid value "'||:new.HISTORICAL_STATUS||'" '||
        'for II_Features.HISTORICAL_STATUS.' );
END IF;
--
IF :new.LOCATION is not null and
:new.LOCATION <> nvl(:old.LOCATION, '@') and
NOT Chk_IiCode( 'FEAT_LOCATION', :new.LOCATION )
THEN
    Raise_Application_Error( -20308,
        'Invalid value "'||:new.LOCATION||'" '||
        'for II_Features.LOCATION.' );
END IF;
--
IF :new.LOCATION_UOM is not null and
:new.LOCATION_UOM <> nvl(:old.LOCATION_UOM, '@') and
NOT Chk_IiCode( 'FEAT_UOM', :new.LOCATION_UOM )
THEN
    Raise_Application_Error( -20309,
        'Invalid value "'||:new.LOCATION_UOM||'" '||
        'for II_Features.LOCATION_UOM.' );
END IF;
--
IF :new.ACCESSIBILITY_STATUS is not null and
:new.ACCESSIBILITY_STATUS <>
nvl(:old.ACCESSIBILITY_STATUS, '@') and
NOT Chk_IiCode( 'ACCESSIBILITY_STATUS', :new.ACCESSIBILITY_STATUS
)
THEN
    Raise_Application_Error( -203010,
        'Invalid value "'||:new.ACCESSIBILITY_STATUS||'" '||
        'for II_Features.ACCESSIBILITY_STATUS.' );
END IF;
--
IF :new.ACQUISITION_METHOD is not null and
:new.ACQUISITION_METHOD <> nvl(:old.ACQUISITION_METHOD, '@') and
NOT Chk_IiCode( 'ACQUISITION_METHOD', :new.ACQUISITION_METHOD )
THEN

```

Oracle Designer Report : ckttrigger

Container: Real Property Management Version :

Table/View Name :II_FEATURES (table)

```

        Raise_Application_Error( -203011,
        'Invalid value "'||:new.ACQUISITION_METHOD||'" '||
        'for II_Features.ACQUISITION_METHOD.' );
END IF;
--
IF :new.OPERATIONAL_STATUS is not null and
:new.OPERATIONAL_STATUS <> nvl(:old.OPERATIONAL_STATUS,'@') and
NOT Chk_IiCode( 'OPERATIONAL_STATUS', :new.OPERATIONAL_STATUS )
THEN
    Raise_Application_Error( -203012,
    'Invalid value "'||:new.OPERATIONAL_STATUS||'" '||
    'for II_Features.OPERATIONAL_STATUS.' );
END IF;
--
IF Inserting
THEN
    -- Supply CN if it is null.
    IF :new.CN is null
    THEN
        :new.CN := Db_Instance.GENERATE_GLOBAL_ID( 'FEAT_CN_SEQ' );
    END IF;
    --
    -- Set the CREATED... audit columns on row insert if not
    supplied.
    IF :new.CREATED_BY is null or
    :new.CREATED_DATE is null or
    :new.CREATED_IN_INSTANCE is null
    THEN
        Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
                                           , :new.CREATED_IN_INSTANCE
                                           , :new.CREATED_DATE );
    END IF;
    ---
ELSIF Updating
THEN
    -- Set the MODIFIED... audit columns on row update.
    Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY
                                      , :new.MODIFIED_IN_INSTANCE
                                      , :new.MODIFIED_DATE );
    ---
END IF;
END;
DECLARE
    vModuleName CONSTANT VARCHAR2(30) := 'II_FEA_RPA_AR_TRG';    --
Added on 01/07/2003
BEGIN
    -- DELETING
    IF DELETING THEN
        -- DELETE from ii_rpa_assets_v_tab
        DELETE FROM ii_rpa_assets_v_tab
        WHERE fea_cn = :old.cn;

    -- INSERTING
    ELSIF INSERTING THEN
        -- Insert into table ii_rpa_assets_v_tab
        BEGIN
            INSERT INTO ii_rpa_assets_v_tab (
                FEA_CN

```

Container: Real Property Management Version :

Table/View Name :II_FEATURES (table)

```

,LU_CN
,PLD_CN
,OBJECT_NAME
,OBJECT_CLASS
,ID
,NAME
,SUB_TYPE
,OWNERSHIP
,ADMIN_ORG
,DEVELOPMENT STATUS
,CREATED_DATE
,CREATED_BY
,SUBLEDGER_FLAG
,subledger_cn
,agency_code
) VALUES (
:new.CN
,NULL
,NULL
,:new.OBJ_NAME
,:new.OBJ_CLASS
,:new.ID
,:new.NAME
,NULL
,:new.OWNERSHIP
,:new.ADMIN_ORG
,:new.DEVELOPMENT STATUS
,:new.CREATED_DATE
,:new.CREATED_BY
,'N'
,null
,:NEW.agency_code -- Added on 10/28/2003 by C.Ma
);

EXCEPTION WHEN OTHERS THEN
    ii_rpa_errors.save_log(vModuleName, 'II_RPA_ASSETS_V_TAB',
:new.cn, null, null, SQLERRM);
END;

-- UPDATING
ELSIF UPDATING THEN
-- Update table ii_rpa_assets_v_tab
UPDATE ii_rpa_assets_v_tab tab
SET (
OBJECT_NAME
,OBJECT_CLASS
,ID
,NAME
,SUB_TYPE
,OWNERSHIP
,ADMIN_ORG
,DEVELOPMENT STATUS
,CREATED_DATE
,CREATED_BY
) = (SELECT
:new.OBJ_NAME OBJECT_NAME
,:new.OBJ_CLASS OBJECT_CLASS
,:new.ID
,:new.NAME

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :II_FEATURES (table)

```

, NULL
, :new.OWNERSHIP
, :new.ADMIN_ORG
, :new.DEVELOPMENT_STATUS
, :new.CREATED_DATE
, :new.CREATED_BY
FROM DUAL)
WHERE tab.fea_cn = :new.cn;

-- Update table ii_rpa subledgers
UPDATE ii_rpa_subledgers sub
SET admin_org = :new.admin_org
WHERE sub.fea_cn = :new.cn
AND ii_sql_func.super_equal(admin_org, :new.admin_org) <>
'TRUE';

-----
-- Audit
-----

DECLARE
  vNewFea ii_features%rowtype;
  vOldFea ii_features%rowtype;
BEGIN
  --vOldFea.GSA_INSTALLATION_CN := :old.GSA_INSTALLATION_CN;--
commented out on 10/28/2003 by C.Ma
  vOldFea.GROSS_SQFT := :old.GROSS_SQFT;
  vOldFea.CN := :old.CN;
  vOldFea.OBJ_TECH := :old.OBJ_TECH;
  vOldFea.OBJ_NAME := :old.OBJ_NAME;
  vOldFea.OBJ_CLASS := :old.OBJ_CLASS;
  vOldFea.ID := :old.ID;
  vOldFea.NAME := :old.NAME;
  vOldFea.SUB_TYPE := :old.SUB_TYPE;
  vOldFea.OWNERSHIP := :old.OWNERSHIP;
  vOldFea.ADMIN_ORG := :old.ADMIN_ORG;
  vOldFea.MASTER_SITE := :old.MASTER_SITE;
  vOldFea.MANAGING_CONT_CN := :old.MANAGING_CONT_CN;
  vOldFea.FUNDING_RESPONSIBILITY := :old.FUNDING_RESPONSIBILITY;
  vOldFea.MAINTENANCE_LEVEL := :old.MAINTENANCE_LEVEL;
  vOldFea.MAINTENANCE_RESPONSIBILITY :=
:old.MAINTENANCE_RESPONSIBILITY;
  vOldFea.PRINCIPAL_MATERIAL := :old.PRINCIPAL_MATERIAL;
  vOldFea.FEATURE_SIZE := :old.FEATURE_SIZE;
  vOldFea.ACQUISITION_METHOD := :old.ACQUISITION_METHOD;
  vOldFea.YEAR_CONSTRUCTED := :old.YEAR_CONSTRUCTED;
  vOldFea.INVENTORIED := :old.INVENTORIED;
  vOldFea.DATE_LAST_INSPECTED := :old.DATE_LAST_INSPECTED;
  vOldFea.CONDITION_RATING := :old.CONDITION_RATING;
  vOldFea.APPRAISAL_RATING := :old.APPRAISAL_RATING;
  vOldFea.DEVELOPMENT_STATUS := :old.DEVELOPMENT_STATUS;
  vOldFea.OPERATIONAL_STATUS := :old.OPERATIONAL_STATUS;
  vOldFea.HISTORICAL_STATUS := :old.HISTORICAL_STATUS;
  vOldFea.ACCESSIBILITY_STATUS := :old.ACCESSIBILITY_STATUS;
  vOldFea.ACCESSIBILITY_REMARKS := :old.ACCESSIBILITY_REMARKS;
  vOldFea.REMARKS := :old.REMARKS;
  vOldFea.CHECKED_OUT_BY := :old.CHECKED_OUT_BY;
  vOldFea.CHECKED_OUT_DATE := :old.CHECKED_OUT_DATE;
  vOldFea.MOST_RECENT_YR_RECONSTR :=
:old.MOST_RECENT_YR_RECONSTR;

```

Oracle Designer Report : cktrgger

Container: Real Property Management

Version :

Table/View Name :II_FEATURES (table)

```

vOldFea.CREATED_BY := :old.CREATED_BY;
vOldFea.CREATED_DATE := :old.CREATED_DATE;
vOldFea.CREATED_IN_INSTANCE := :old.CREATED_IN_INSTANCE;
vOldFea.MODIFIED_BY := :old.MODIFIED_BY;
vOldFea.MODIFIED_DATE := :old.MODIFIED_DATE;
vOldFea.MODIFIED_IN_INSTANCE := :old.MODIFIED_IN_INSTANCE;
--vNewFea.GSA_INSTALLATION_CN := :new.GSA_INSTALLATION_CN;--
commented out on 10/28/2003 by C.Ma
vNewFea.GROSS_SQFT := :new.GROSS_SQFT;
vNewFea.CN := :new.CN;
vNewFea.OBJ_TECH := :new.OBJ_TECH;
vNewFea.OBJ_NAME := :new.OBJ_NAME;
vNewFea.OBJ_CLASS := :new.OBJ_CLASS;
vNewFea.ID := :new.ID;
vNewFea.NAME := :new.NAME;
vNewFea.SUB_TYPE := :new.SUB_TYPE;
vNewFea.OWNERSHIP := :new.OWNERSHIP;
vNewFea.ADMIN_ORG := :new.ADMIN_ORG;
vNewFea.MASTER_SITE := :new.MASTER_SITE;
vNewFea.MANAGING_CONT_CN := :new.MANAGING_CONT_CN;
vNewFea.FUNDING_RESPONSIBILITY := :new.FUNDING_RESPONSIBILITY;
vNewFea.MAINTENANCE_LEVEL := :new.MAINTENANCE_LEVEL;
vNewFea.MAINTENANCE_RESPONSIBILITY :=
:new.MAINTENANCE_RESPONSIBILITY;
vNewFea.PRINCIPAL_MATERIAL := :new.PRINCIPAL_MATERIAL;
vNewFea.FEATURE_SIZE := :new.FEATURE_SIZE;
vNewFea.ACQUISITION_METHOD := :new.ACQUISITION_METHOD;
vNewFea.YEAR_CONSTRUCTED := :new.YEAR_CONSTRUCTED;
vNewFea.INVENTORIED := :new.INVENTORIED;
vNewFea.DATE_LAST_INSPECTED := :new.DATE_LAST_INSPECTED;
vNewFea.CONDITION_RATING := :new.CONDITION_RATING;
vNewFea.APPRAISAL_RATING := :new.APPRAISAL_RATING;
vNewFea.DEVELOPMENT_STATUS := :new.DEVELOPMENT_STATUS;
vNewFea.OPERATIONAL_STATUS := :new.OPERATIONAL_STATUS;
vNewFea.HISTORICAL_STATUS := :new.HISTORICAL_STATUS;
vNewFea.ACCESSIBILITY_STATUS := :new.ACCESSIBILITY_STATUS;
vNewFea.ACCESSIBILITY_REMARKS := :new.ACCESSIBILITY_REMARKS;
vNewFea.REMARKS := :new.REMARKS;
vNewFea.CHECKED_OUT_BY := :new.CHECKED_OUT_BY;
vNewFea.CHECKED_OUT_DATE := :new.CHECKED_OUT_DATE;
vNewFea.MOST_RECENT_YR_RECONSTR :=
:new.MOST_RECENT_YR_RECONSTR;
vNewFea.CREATED_BY := :new.CREATED_BY;
vNewFea.CREATED_DATE := :new.CREATED_DATE;
vNewFea.CREATED_IN_INSTANCE := :new.CREATED_IN_INSTANCE;
vNewFea.MODIFIED_BY := :new.MODIFIED_BY;
vNewFea.MODIFIED_DATE := :new.MODIFIED_DATE;
vNewFea.MODIFIED_IN_INSTANCE := :new.MODIFIED_IN_INSTANCE;

ii_rpa auditing.audit_fea('U', :new.modified_by,
:new.modified_date, vOldFea, vNewFea);
END;

END IF;

END;
```

Oracle Designer Report : ckttrigger

Container: Real Property Management Version :

Table/View Name :ACCPISHMENT_INSTRUMENTS (table)

Trigger Name :	AI_BRIU_TRG	Create?	Yes
PL/SQL Module			
Name :	AI_BRIU_TRG	Enabled?	Yes

PL/SQL File :

Purpose : .

Trigger Time:	BEFORE	Level:	ROW	Insert?	Yes
---------------	--------	--------	-----	---------	-----

Delete? No

New Alias	NEW	Update?	Yes
-----------	-----	---------	-----

Old Alias	: OLD
-----------	-------

Trigger Logic

```

DECLARE
BEGIN
  IF inserting
  THEN
    IF :new.cn is null
    THEN :new.cn :=
db_instance.generate_global_id('ACCINST_CN_SEQ');
    END IF;
    db_instance.insert_audit_columns(:new.created_by,
: new.created_in instance,
: new.created_date);
    ELSE db_instance.update_audit_columns(:new.modified_by,
: new.modified_in instance,
: new.modified_date);
    END IF;
END;

```

Container: Real Property Management

Version :

Table/View Name :ACCPHISHMENT_INSTRUMENTS (table)

Trigger Name : II_AI_ARIUD_TRG Create? Yes

PL/SQL Module

Name : II_AI_ARIUD_TRG Enabled? Yes

PL/SQL File :

Purpose : After-row I/U/D to populate range permit audit tables.

Trigger Time: AFTER Level: ROW Insert? Yes

Delete? Yes

New Alias Update? Yes

Old Alias :

Trigger Logic

```

DECLARE
BEGIN
  IF inserting
  THEN
    IF :new.cn is null
    THEN :new.cn :=
db_instance.generate_global_id('ACCINST_CN_SEQ');
    END IF;
    db_instance.insert_audit_columns(:new.created_by,
:~new.created_in_instance,
:~new.created_date);
    ELSE db_instance.update_audit_columns(:new.modified_by,
:~new.modified_in_instance,
:~new.modified_date);
  END IF;
END;
DECLARE
BEGIN
  IF inserting
  THEN
    IF :new.cn is null
    THEN :new.cn :=
db_instance.generate_global_id('ACCINST_CN_SEQ');
    END IF;
    db_instance.insert_audit_columns(:new.created_by,
:~new.created_in_instance,
:~new.created_date);
    ELSE db_instance.update_audit_columns(:new.modified_by,
:~new.modified_in_instance,
:~new.modified_date);
  END IF;
END;
DECLARE
BEGIN
  IF inserting
  THEN
    IF :new.cn is null
    THEN :new.cn :=
db_instance.generate_global_id('ACCINST_CN_SEQ');
    END IF;
    db_instance.insert_audit_columns(:new.created_by,

```

Oracle Designer Report : cktrigger

Container: Real Property Management Version :

Table/View Name :ACCPHISHMENT_INSTRUMENTS (table)

```

        :new.created_in_instance,
        :new.created_date);
    ELSE db_instance.update_audit_columns(:new.modified_by,
        :new.modified_in_instance,
        :new.modified_date);
    END IF;
END;
Declare
--
    VRNGPAUDRecord ii_range_permit_audits%rowtype;
--
Begin
--
    if INSERTING then
--
        if :new.obj_name = 'RANGE_PERMIT' then
--
            if ii_range_permits_audit.vAuditCN is not null then -- set in
instead-of trigger of II_RANGE_PERMITS_V
                null; -- Audit record already written.
            else
                VRNGPAUDRecord.changed_by           := user;
                VRNGPAUDRecord.changed_on           := sysdate;
                VRNGPAUDRecord.change_type           := 'I';
                VRNGPAUDRecord.old_id                := null;
                VRNGPAUDRecord.new_id               := :new.ID;
                VRNGPAUDRecord.old_cn               := null;
                VRNGPAUDRecord.new_cn               := :new.CN;
                VRNGPAUDRecord.old_description       := null;
                VRNGPAUDRecord.new_description       := :new.Description;
                VRNGPAUDRecord.old_managing_cont_cn := null;
                VRNGPAUDRecord.new_managing_cont_cn :=
:new.managing_cont_cn;
                VRNGPAUDRecord.old_obj_name         := null;
                VRNGPAUDRecord.new_obj_name         := :new.obj_name;
                VRNGPAUDRecord.old_permittee_cn      := null;
                VRNGPAUDRecord.new_permittee_cn      := null;
                VRNGPAUDRecord.old_supercedes_rngpmt_cn := null;
                VRNGPAUDRecord.new_supercedes_rngpmt_cn := null;
                VRNGPAUDRecord.old_split_bill_flag   := null;
                VRNGPAUDRecord.new_split_bill_flag   := null;
                VRNGPAUDRecord.old_split_bill_month  := null;
                VRNGPAUDRecord.new_split_bill_month  := null;
                VRNGPAUDRecord.old_status            := null;
                VRNGPAUDRecord.new_status            := null;
                VRNGPAUDRecord.old_escrow_waiver_flag := null;
                VRNGPAUDRecord.new_escrow_waiver_flag := null;
                VRNGPAUDRecord.old_bid_bill_flag     := null;
                VRNGPAUDRecord.new_bid_bill_flag     := null;
                VRNGPAUDRecord.change_occ_in_table :=
'ACCPHISHMENT_INSTRUMENTS';
                ii_range_permits_audit.Insert_II_RANGE_PERMIT_AUDITS (
VRNGPAUDRecord );
            end if;
--
        end if;
--
    elsif UPDATING then

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :ACCPISHMENT_INSTRUMENTS (table)

```

--
if :new.obj_name = 'RANGE_PERMIT' then
  if nvl(:new.id,'@') != nvl(:old.id,'@') or
     nvl(:new.description,'@') != nvl(:old.description,'@')
or
  nvl(:new.managing_cont_cn,'@') !=
nvl(:old.managing_cont_cn,'@') or
  nvl(:new.obj_name,'@') != nvl(:old.obj_name,'@') or
  nvl(:new.cn,'@') != nvl(:old.cn,'@') or
  nvl(:new.description,'@') != nvl(:old.description,'@')
then
  if ii_range_permits_audit.vAuditCN is not null then -- set in
instead-of trigger of II_RANGE_PERMITS V
  null; -- Audit record already written.
  else
    VRNGPAUDRecord.changed_by := user;
    VRNGPAUDRecord.changed_on := sysdate;
    VRNGPAUDRecord.change_type := 'U';
    VRNGPAUDRecord.old_id := :old.ID;
    VRNGPAUDRecord.new_id := :new.ID;
    VRNGPAUDRecord.old_cn := :old.CN;
    VRNGPAUDRecord.new_cn := :new.CN;
    VRNGPAUDRecord.old_description := :old.Description;
    VRNGPAUDRecord.new_description := :new.Description;
    VRNGPAUDRecord.old_managing_cont_cn :=
:old.managing_cont_cn;
    VRNGPAUDRecord.new_managing_cont_cn :=
:new.managing_cont_cn;
    VRNGPAUDRecord.old_obj_name := :old.obj_name;
    VRNGPAUDRecord.new_obj_name := :new.obj_name;
    VRNGPAUDRecord.old_permittee_cn := null;
    VRNGPAUDRecord.new_permittee_cn := null;
    VRNGPAUDRecord.old_supercedes_rngpmt_cn := null;
    VRNGPAUDRecord.new_supercedes_rngpmt_cn := null;
    VRNGPAUDRecord.old_split_bill_flag := null;
    VRNGPAUDRecord.new_split_bill_flag := null;
    VRNGPAUDRecord.old_split_bill_month := null;
    VRNGPAUDRecord.new_split_bill_month := null;
    VRNGPAUDRecord.old_status := null;
    VRNGPAUDRecord.new_status := null;
    VRNGPAUDRecord.old_escrow_waiver_flag := null;
    VRNGPAUDRecord.new_escrow_waiver_flag := null;
    VRNGPAUDRecord.old_bid_bill_flag := null;
    VRNGPAUDRecord.new_bid_bill_flag := null;
    VRNGPAUDRecord.change_occ_in_table :=
'ACCPISHMENT_INSTRUMENTS';
    ii_range_permits_audit.Insert_II_RANGE_PERMIT_AUDITS (
VRNGPAUDRecord );
  end if;
end if;
end if;
--
elsif DELETING then
  --
  if :old.obj_name = 'RANGE_PERMIT' then
    if ii_range_permits_audit.vAuditCN is not null then -- set in
instead-of trigger of II_RANGE_PERMITS V
    null; -- Audit record already written.
  else

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :ACCPISHMENT_INSTRUMENTS (table)

```

VRNGPAUDRecord.changed_by           := user;
VRNGPAUDRecord.changed_on           := sysdate;
VRNGPAUDRecord.change_type           := 'D';
VRNGPAUDRecord.old_id                := :old.ID;
VRNGPAUDRecord.new_id                := null;
VRNGPAUDRecord.old_cn                := :old.CN;
VRNGPAUDRecord.new_cn                := null;
VRNGPAUDRecord.old_description       := :old.Description;
VRNGPAUDRecord.new_description       := null;
VRNGPAUDRecord.old_managing_cont_cn :=
:old.managing_cont_cn;
VRNGPAUDRecord.new_managing_cont_cn := null;
VRNGPAUDRecord.old_obj_name          := :old.obj_name;
VRNGPAUDRecord.new_obj_name          := null;
VRNGPAUDRecord.old_permittee_cn      := null;
VRNGPAUDRecord.new_permittee_cn      := null;
VRNGPAUDRecord.old_supercedes_rngpmt_cn := null;
VRNGPAUDRecord.new_supercedes_rngpmt_cn := null;
VRNGPAUDRecord.old_split_bill_flag  := null;
VRNGPAUDRecord.new_split_bill_flag  := null;
VRNGPAUDRecord.old_split_bill_month := null;
VRNGPAUDRecord.new_split_bill_month := null;
VRNGPAUDRecord.old_status            := null;
VRNGPAUDRecord.new_status            := null;
VRNGPAUDRecord.old_escrow_waiver_flag := null;
VRNGPAUDRecord.new_escrow_waiver_flag := null;
VRNGPAUDRecord.old_bid_bill_flag     := null;
VRNGPAUDRecord.new_bid_bill_flag     := null;
VRNGPAUDRecord.change_occ_in_table :=
'ACCPISHMENT_INSTRUMENTS';
    ii_range_permits_audit.Insert_II_RANGE_PERMIT_AUDITS (
VRNGPAUDRecord );
    end if;
    end if;
    end if;
End;
```

Container: Real Property Management

Version :

Table/View Name :II_GA_COOPERATORS (table)

Trigger Name :	T_RBIU_COOPERATOR	Create?	Yes
PL/SQL Module Name :	T_RBIU_COOPERATOR	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
		Delete?	No
New Alias :	NEW	Update?	Yes
Old Alias :	OLD		

Trigger Logic

```

DECLARE
BEGIN
  BEGIN
    IF inserting
    THEN
      -- Create the CN if not supplied on row insert.
      IF :new.CN is null
      THEN
        :new.CN := Db_Instance.GENERATE_GLOBAL_ID( 'II_GA_SEQ' );
      END IF;
      --
      -- Set the CREATED... audit columns on row insert if not
      supplied.
      IF :new.CREATED_BY is null or
        :new.CREATED_DATE is null or
        :new.CREATED_IN_INSTANCE is null
      THEN
        Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
                                          , :new.CREATED_IN_INSTANCE
                                          , :new.CREATED_DATE );
      END IF;
      ---
    ELSIF updating
    THEN
      -- Set the MODIFIED... audit columns on row update.
      Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY
                                       , :new.MODIFIED_IN_INSTANCE
                                       , :new.MODIFIED_DATE );
      ---
    END IF;
  END;
END;

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :II_BLDG_FLOOR_USES (table)

Trigger Name : BLDFLUSE_IICODE_TRG Create? Yes

PL/SQL Module

Name : BLDFLUSE_IICODE_TRG Enabled? Yes

PL/SQL File :

Purpose : .

Trigger Time: BEFORE Level: ROW Insert? Yes

Delete? No

New Alias NEW Update? Yes

Old Alias : OLD

Trigger Logic

DECLARE

BEGIN

```

    if (:new.SPACE_USAGE_TYPE is not null
        and not chk_iicode('SPACE_USAGE_TYPE',:new.SPACE_USAGE_TYPE))
    then
        raise_application_error(-20300, 'Invalid value; II_CODES name not
found.');
```

```

    end if;
```

END;

Container: Real Property Management Version :

Table/View Name :II_DOCUMENTATIONS (table)

Trigger Name : II_TIU_DOCMNT Create? Yes

PL/SQL Module

Name : II_TIU_DOCMNT Enabled? Yes

PL/SQL File :

Purpose : .

Trigger Time: BEFORE Level: ROW Insert? Yes

Delete? No

New Alias NEW Update? Yes

Old Alias : OLD

Trigger Logic

DECLARE

BEGIN

IF INSERTING THEN

IF :new.cn IS NULL THEN

:new.cn :=

db_instance.generate_global_id('II_DOCMNT_CN_SEQ');

END IF;

db_instance.insert_audit_columns

(:new.created_by, :new.created_in_instance, :new.created_date);

ELSE

db_instance.update_audit_columns

(:new.modified_by, :new.modified_in_instance, :new.modified_date);

END IF;

END;

Container: Real Property Management Version :

Table/View Name :II_DOCUMENTATIONS (table)

Trigger Name : II_TIU_DOCMNT_IICODE_CHK Create? Ye.
 PL/SQL Module
 Name : II_TIU_DOCMNT_IICODE_CHK Enabled? Ye.
 PL/SQL File :
 Purpose : .
 Trigger Time: BEFORE Level: ROW Insert? Ye.
 Delete? No
 New Alias NEW Update? Ye.
 Old Alias : OLD

Trigger Logic

```

DECLARE
BEGIN
  IF (:new.SUBTYPE IS NOT NULL
    AND NOT chk_iicode(:new.OBJ_NAME||'_SUB_TYPE',:new.SUBTYPE))
  then
    raise_application_error(-20301, Initcap(:new.OBJ_NAME)||
      ' SuBtype not found - Please re-enter');
  END IF;
  IF nvl(:new.SUBTYPE,'@') = 'NEPA DECISION DOCUMENT' then
    IF nvl(:new.SUBTYPE_CATEGORY,'@') !=
nvl(:old.SUBTYPE_CATEGORY,'@') and
      NOT chk_iicode('DOCUMENT_NEPA_DECISION_DOCUMENT_CATEGORY',
:new.SUBTYPE_CATEGORY) THEN
      raise_application_error(-20303,'NEPA Decision Document
Category '||
        :new.SUBTYPE_CATEGORY||' not found - Please re-
enter');
    END IF;
  ELSE
    IF (:new.SUBTYPE_CATEGORY IS NOT NULL
      AND nvl(:new.SUBTYPE_CATEGORY,'@') !=
nvl(:old.SUBTYPE_CATEGORY,'@')
      AND NOT chk_iicode(:new.OBJ_NAME||'_'||
:new.SUBTYPE||'_SUBTYPE_CATEGORY',:new.SUBTYPE_CATEGORY)) then
      raise_application_error(-20302,
nvl(Initcap(replace(:new.SUBTYPE,'_',' ')), 'Subtype')|| '
Category '||
        'not found - Please re-enter');
    END IF;
  END IF;
END;

```

Oracle Designer Report : ckttrigger

Container: Real Property Management

Version :

Table/View Name :II_POOLED (table)

Trigger Name :	II_POOLED_CN_AUDIT	Create?	Yes
PL/SQL Module			
Name :	II_POOLED_CN_AUDIT	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
New Alias	NEW	Delete?	No
Old Alias :	OLD	Update?	Yes

Trigger Logic

```

DECLARE
BEGIN
    IF (:new.SUBTYPE IS NOT NULL
        AND NOT chk_iicode(:new.OBJ_NAME||'_SUB_TYPE',:new.SUBTYPE))
    then
        raise_application_error(-20301, Initcap(:new.OBJ_NAME)||
            ' SuBtype not found - Please re-enter');
    END IF;
    IF nvl(:new.SUBTYPE,'@') = 'NEPA DECISION DOCUMENT' then
        IF nvl(:new.SUBTYPE_CATEGORY,'@') !=
nvl(:old.SUBTYPE_CATEGORY,'@') and
            NOT chk_iicode('DOCUMENT_NEPA_DECISION_DOCUMENT_CATEGORY',
:new.SUBTYPE_CATEGORY) THEN
            raise_application_error(-20303,'NEPA Decision Document
Category '||
:new.SUBTYPE_CATEGORY||' not found - Please re-
enter');
        END IF;
    ELSE
        IF (:new.SUBTYPE_CATEGORY IS NOT NULL
            AND nvl(:new.SUBTYPE_CATEGORY,'@') !=
nvl(:old.SUBTYPE_CATEGORY,'@')
            AND NOT chk_iicode(:new.OBJ_NAME||'_'||
:new.SUBTYPE||'_SUBTYPE_CATEGORY',:new.SUBTYPE_CATEGORY)) then
            raise_application_error(-20302,
nvl(Initcap(replace(:new.SUBTYPE,'_',' ')), 'Subtype')|| '
Category '||
            'not found - Please re-enter');
        END IF;
    END IF;
END;
DECLARE
--
--
v_tmp_hold CHAR(1);
--
--
v_admin_org_id II_CONTACTS.ID%TYPE;
--
--

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :II_POOLEDS (table)

```

p_id II_POOLEDS.ID%TYPE;
--
--
p_admin_org_cn II_POOLEDS.ADMIN_ORG_CN%TYPE;
--
--
p_obj_name II_POOLEDS.OBJ_NAME%TYPE;
--
--
admin_org_missing EXCEPTION;
PRAGMA EXCEPTION_INIT(admin_org_missing, -20004);
--
--
id_missing EXCEPTION;
PRAGMA EXCEPTION_INIT(id_missing, -20003);
--
--
obj_name_missing EXCEPTION;
PRAGMA EXCEPTION_INIT(obj_name_missing, -20002);
--
--
contact_notfound EXCEPTION;
PRAGMA EXCEPTION_INIT(contact_notfound, -20000);
--
--
duplicate_found EXCEPTION;
PRAGMA EXCEPTION_INIT(duplicate_found, -20001);
--
-- POOLED INSTANCE EXISTS - DUPLICATE
CURSOR POOLED_EXIST_CUR(
p_obj_name ii_pooled.obj_name%type ,
p_id ii_pooled.id%type ,
p_contact_id ii_contacts.id%type )
IS
SELECT '1'
FROM II_POOLEDS p, ii_contacts cnt
WHERE substr(cnt.id,1,4) = p_contact_id
AND p.admin_org_cn = cnt.cn
AND p.OBJ_NAME = P_OBJ_NAME
AND p.ID = P_ID;
--
-- FETCH CONTACT ID FOR ADMIN ORG CN.
CURSOR CONTACT_ID_CUR(
search_admin_org_cn ii_contacts.cn%type )
IS
SELECT id
FROM ii_contacts
WHERE cn = search_admin_org_cn;
BEGIN
BEGIN
IF INSERTING THEN
if :new.obj_name is null then
-- raise obj_name_missing exception;
raise_application_error(-20002,
'Pooled asset Type is mandatory.');
```

Oracle Designer Report : cktggr

Container: Real Property Management Version :

Table/View Name :II_POOLED (table)

```

        -- raise id_missing exception;
        raise_application_error(-20003,
            'Pooled asset ID is mandatory.');
```

elseif :new.admin_org_cn is null then

```

        -- raise admin_org_missing exception;
        raise_application_error(-20004,
            'Pooled asset Admin Org is mandatory.');
```

end if;

```

    -- initialize instance values for cursors

    p_id := :new.id;
    p_obj_name := :new.obj_name;
    p_admin_org_cn := :new.admin_org_cn;

    open contact_id_cur(p_admin_org_cn);
    fetch contact_id_cur into v_admin_org_id;
    if contact_id_cur%notfound then
        close contact_id_cur;

        -- raise contact_notfound exception;
        raise_application_error(-20000,
            'Contact ID was not found.');
```

else

```

    -- initialize
    v_admin_org_id := substr(v_admin_org_id,1,4);
    close contact_id_cur;

end if;
```

-- check for duplicates

```

OPEN pooled_exist_cur(p_obj_name, p_id, v_admin_org_id);
FETCH pooled_exist_cur INTO v_tmp_hold;

IF pooled_exist_cur%FOUND THEN

    CLOSE pooled_exist_cur;
    -- raise duplicate found exception;
    raise_application_error(-20001,
        'Duplicate Pooled Asset Found with Type '||
        p_obj_name||', ID '||p_id||', and Contact ID '||
        v_admin_org_id||'.');
```

ELSE

```

    /* Instance does not exist */
    CLOSE pooled_exist_cur;
END IF;
```

IF :new.cn IS NULL THEN

```

    :new.cn :=
db_instance.generate_global_id('II_POOLED_CN_SEQ');
```

Oracle Designer Report : cktgger

Container: Real Property Management Version :

Table/View Name :II_POOLEDS (table)

```

        END IF;
        db_instance.insert_audit_columns
            (:new.created_by, :new.created_in_instance,
:new.created_date);
        ELSE
            db_instance.update_audit_columns
                (:new.modified_by, :new.modified_in_instance,
:new.modified_date);
        END IF;
    END;
EXCEPTION
    WHEN duplicate_found THEN
        RAISE;
    WHEN admin_org_missing THEN
        RAISE;
    WHEN id_missing THEN
        RAISE;
    WHEN obj_name_missing THEN
        RAISE;
    WHEN contact_notfound THEN
        RAISE;
END;
```

Container: Real Property Management

Version :

Table/View Name :II_POOLEDS (table)

Trigger Name :	II_POOLED_RPA_AR_TRG	Create?
PL/SQL Module		
Name :	II_POOLED_RPA_AR	Enabled
PL/SQL File :		
Purpose :	Maintain RPA functions when AFTER ROW	
Trigger Time:	AFTER	Level: ROW
		Insert?
		Delete?
New Alias		Update?
Old Alias :		

Trigger Logic

```

DECLARE
--
--
v_tmp_hold CHAR(1);
--
--
v_admin_org_id II_CONTACTS.ID%TYPE;
--
--
p_id II_POOLEDS.ID%TYPE;
--
--
p_admin_org_cn II_POOLEDS.ADMIN_ORG_CN%TYPE;
--
--
p_obj_name II_POOLEDS.OBJ_NAME%TYPE;
--
--
admin_org_missing EXCEPTION;
PRAGMA EXCEPTION_INIT(admin_org_missing, -20004);
--
--
id_missing EXCEPTION;
PRAGMA EXCEPTION_INIT(id_missing, -20003);
--
--
obj_name_missing EXCEPTION;
PRAGMA EXCEPTION_INIT(obj_name_missing, -20002);
--
--
contact_notfound EXCEPTION;
PRAGMA EXCEPTION_INIT(contact_notfound, -20000);
--
--
duplicate_found EXCEPTION;
PRAGMA EXCEPTION_INIT(duplicate_found, -20001);
--
-- POOLED INSTANCE EXISTS - DUPLICATE
CURSOR POOLED_EXIST_CUR(
p_obj_name ii_pooled.obj_name%type ,
p_id ii_pooled.id%type ,
p_contact_id ii_contacts.id%type )

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :II_POOLED5 (table)

```

IS
SELECT '1'
FROM   II_POOLED5 p, ii_contacts cnt
WHERE  substr(cnt.id,1,4) = p_contact_id
AND    p.admin_org_cn = cnt.cn
AND    p.OBJ_NAME = P_OBJ_NAME
AND    p.ID = P_ID;
--
-- FETCH CONTACT ID FOR ADMIN ORG CN.
CURSOR CONTACT_ID_CUR(
search_admin_org_cn   ii_contacts.cn%type )
IS
SELECT id
FROM   ii_contacts
WHERE  cn = search_admin_org_cn;
BEGIN
  IF INSERTING THEN
    if :new.obj_name is null then

      -- raise obj_name missing exception;
      raise_application_error(-20002,
        'Pooled asset Type is mandatory.');
```

elseif :new.id is null then

```

      -- raise id missing exception;
      raise_application_error(-20003,
        'Pooled asset ID is mandatory.');
```

elseif :new.admin_org_cn is null then

```

      -- raise admin_org missing exception;
      raise_application_error(-20004,
        'Pooled asset Admin Org is mandatory.');
```

end if;

```

-- initialize instance values for cursors

p_id := :new.id;
p_obj_name := :new.obj_name;
p_admin_org_cn := :new.admin_org_cn;

open contact_id_cur(p_admin_org_cn);
fetch contact_id_cur into v_admin_org_id;
if contact_id_cur%notfound then
  close contact_id_cur;

  -- raise contact_notfound exception;
  raise_application_error(-20000,
    'Contact ID was not found.');
```

else

```

  -- initialize
  v_admin_org_id := substr(v_admin_org_id,1,4);
  close contact_id_cur;
```

Oracle Designer Report : cktggr

Container: Real Property Management Version :

Table/View Name :II_POOLED5 (table)

```

end if;

-- check for duplicates

OPEN pooled_exist_cur(p_obj_name, p_id, v_admin_org_id);
FETCH pooled_exist_cur INTO v_tmp_hold;

IF pooled_exist_cur%FOUND THEN

    CLOSE pooled_exist_cur;
    -- raise duplicate found exception;
    raise_application_error(-20001,
        'Duplicate Pooled Asset Found with Type '||
        p_obj_name||', ID '||p_id||', and Contact ID '||
        v_admin_org_id||'.');

ELSE
    /* Instance does not exist */
    CLOSE pooled_exist_cur;
END IF;

IF :new.cn IS NULL THEN
    :new.cn :=
db_instance.generate_global_id('II_POOLED_CN_SEQ');
END IF;
db_instance.insert_audit_columns
(:new.created_by, :new.created_in_instance,
:new.created_date);
ELSE
    db_instance.update_audit_columns
(:new.modified_by, :new.modified_in_instance,
:new.modified_date);
END IF;
END;
EXCEPTION
WHEN duplicate_found THEN
RAISE;
WHEN admin_org_missing THEN
RAISE;
WHEN id_missing THEN
RAISE;
WHEN obj_name_missing THEN
RAISE;
WHEN contact_notfound THEN
RAISE;

END;
-- DELETING
IF DELETING THEN
    DELETE FROM ii_rpa_assets_v_tab
    WHERE
        pld_cn = :OLD.cn;

-- INSERTING
ELSIF INSERTING THEN
    INSERT INTO ii_rpa_assets_v_tab (

```

Oracle Designer Report : cktrgger

Container: Real Property Management Version :

Table/View Name :II_POOLED5 (table)

```

        FEA_CN
        , LU_CN
        , PLD_CN
        , OBJECT_NAME
        , OBJECT_CLASS
        , ID
        , NAME
        , SUB_TYPE
        , OWNERSHIP
        , ADMIN_ORG
        , DEVELOPMENT STATUS
        , CREATED_DATE
        , CREATED_BY
        , SUBLEDGER_FLAG
        , subledger_cn
        , agency_code
    ) VALUES (
        NULL
        , NULL
        , :NEW.CN
        , :NEW.OBJ_NAME
        , :NEW.OBJ_CLASS
        , :NEW.ID
        , :NEW.NAME
        , NULL
        , NULL
        , :NEW.ADMIN_ORG
        , NULL
        , :NEW.CREATED_DATE
        , :NEW.CREATED_BY
        , 'N'
        , null
        , :NEW.agency_code -- Added on 10/28/2003 by C.Ma
    );

-- UPDATING
ELSIF UPDATING THEN
-- Update table ii_rpa_assets_v_tab
UPDATE ii_rpa_assets_v_tab
SET OBJECT_NAME = :NEW.OBJ_NAME
, OBJECT_CLASS = :NEW.OBJ_CLASS
, ID = :NEW.ID
, NAME = :NEW.NAME
, ADMIN_ORG = :NEW.ADMIN_ORG
WHERE
    pld_cn = :NEW.cn;

-- Update table ii_rpa_subledgers
UPDATE ii_rpa_subledgers sub
SET admin_org = :new.admin_org
WHERE sub.pld_cn = :new.cn
AND ii_sql_func.super_equal(admin_org, :new.admin_org) <>
'TRUE';

-----
-- Audit
-----
DECLARE
    vNewPld ii_pooled5%rowtype;

```

Oracle Designer Report : cktgrger

Container: Real Property Management Version :

Table/View Name :II_POOLED5 (table)

```

        vOldPld ii_pooled5%rowtype;
    BEGIN
        --vOldPld.GSA_INSTALLATION_CN := :old.GSA_INSTALLATION_CN;--
        commented out on 10/28/2003 by C.Ma
        vOldPld.CN := :old.CN;
        vOldPld.OBJ_CLASS := :old.OBJ_CLASS;
        vOldPld.OBJ_TECH := :old.OBJ_TECH;
        vOldPld.OBJ_NAME := :old.OBJ_NAME;
        vOldPld.ID := :old.ID;
        vOldPld.ADMIN_ORG := :old.ADMIN_ORG;
        vOldPld.NAME := :old.NAME;
        vOldPld.MASTER_SITE := :old.MASTER_SITE;
        vOldPld.REMARKS := :old.REMARKS;
        vOldPld.CREATED_BY := :old.CREATED_BY;
        vOldPld.CREATED_DATE := :old.CREATED_DATE;
        vOldPld.CREATED_IN_INSTANCE := :old.CREATED_IN_INSTANCE;
        vOldPld.MODIFIED_BY := :old.MODIFIED_BY;
        vOldPld.MODIFIED_DATE := :old.MODIFIED_DATE;
        vOldPld.MODIFIED_IN_INSTANCE := :old.MODIFIED_IN_INSTANCE;

        --vNewPld.GSA_INSTALLATION_CN := :new.GSA_INSTALLATION_CN;--
        commented out on 10/28/2003 by C.Ma
        vNewPld.CN := :new.CN;
        vNewPld.OBJ_CLASS := :new.OBJ_CLASS;
        vNewPld.OBJ_TECH := :new.OBJ_TECH;
        vNewPld.OBJ_NAME := :new.OBJ_NAME;
        vNewPld.ID := :new.ID;
        vNewPld.ADMIN_ORG := :new.ADMIN_ORG;
        vNewPld.NAME := :new.NAME;
        vNewPld.MASTER_SITE := :new.MASTER_SITE;
        vNewPld.REMARKS := :new.REMARKS;
        vNewPld.CREATED_BY := :new.CREATED_BY;
        vNewPld.CREATED_DATE := :new.CREATED_DATE;
        vNewPld.CREATED_IN_INSTANCE := :new.CREATED_IN_INSTANCE;
        vNewPld.MODIFIED_BY := :new.MODIFIED_BY;
        vNewPld.MODIFIED_DATE := :new.MODIFIED_DATE;
        vNewPld.MODIFIED_IN_INSTANCE := :new.MODIFIED_IN_INSTANCE;

        ii_rpa auditing.audit_pld('U', :new.modified_by,
        :new.modified_date, vOldPld, vNewPld);
    END;
END IF;
```

Container: Real Property Management Version :

Table/View Name :II_GSA_FILE_LOGS (table)

Trigger Name :	II_GSA_FILE_LOGS_BRI	Create?	Yes
PL/SQL Module Name :	II_GSA_FILE_LOGS_BRIU_TRG	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
New Alias	NEW	Delete?	No
Old Alias	: OLD	Update?	Yes

Trigger Logic

```

DECLARE
BEGIN
  IF inserting AND :new.cn IS NULL
  THEN
    :new.cn := db_instance.generate_global_id('II_GSA_CN_SEQ');
  END IF;
END;
```


Container: Real Property Management Version :

Table/View Name :II_GSA_FILE_LOGS (table)

Trigger Name : II_GSA_FILE_LOGS_BRI Create? Yes

PL/SQL Module

Name : II_GSA_FILE_LOGS_BRIU_TRG Enabled? Yes

PL/SQL File :

Purpose : .

Trigger Time: BEFORE Level: ROW Insert? Yes

Delete? No

New Alias NEW Update? Yes

Old Alias : OLD

Trigger Logic

```
DECLARE
BEGIN
  IF inserting AND :new.cn IS NULL
  THEN
    :new.cn := db_instance.generate_global_id('II_GSA_CN_SEQ');
  END IF;
END;
```

Container: Real Property Management Version :

Table/View Name :II_AGENCY_CERTIFICATION (table)

Trigger Name :	AGCY_CERT_AUDIT	Create?	Yes
PL/SQL Module			
Name :	AGCY_CERT_AUDIT	Enabled?	Yes
PL/SQL File :			
Purpose :	.		
Trigger Time:	BEFORE	Level:	ROW
		Insert?	Yes
New Alias		Delete?	No
Old Alias :		Update?	Yes

Trigger Logic

```

BEGIN
  IF inserting THEN
    :new.created_by:=user;
    :new.created_date:=sysdate;
  ELSE
    :new.modified_by:=user;
    :new.modified_date:=sysdate;
  END IF;
END;
```

Container: Real Property Management Version :

Table/View Name :II_FEA_PLANNING_STATUS (table)

Trigger Name : II_RBIU_FEA_PLAN_STATUS Create? Yes

PL/SQL Module

Name : II_RBIU_FEA_PLAN_STATUS Enabled? Yes

PL/SQL File :

Purpose : .

Trigger Time: BEFORE Level: ROW Insert? Yes

Delete? Yes

New Alias Update? Yes

Old Alias :

Trigger Logic

```

DECLARE
BEGIN
  -- Set the CREATED... audit columns on row insert if not supplied.
  IF :new.CREATED_BY is null or
     :new.CREATED_DATE is null or
     :new.CREATED_IN_INSTANCE is null
  THEN
    Db_Instance.INSERT_AUDIT_COLUMNS( :new.CREATED_BY
                                     , :new.CREATED_IN_INSTANCE
                                     , :new.CREATED_DATE );
  END IF;
  ---
  ELSIF Updating
  THEN
    -- Set the MODIFIED... audit columns on row update.
    Db_Instance.UPDATE_AUDIT_COLUMNS( :new.MODIFIED_BY
                                     , :new.MODIFIED_IN_INSTANCE
                                     , :new.MODIFIED_DATE );
    ---
  END IF;
END;
```

Oracle Designer
DATABASE TRIGGER REPORT
End of Report