

Actions x Reports x

.sql x cse40 x ALLOCREGISTER_ALLOCQTY_TRG x ALLOCREGISTER_ALOCYEAR_TRG x ALLOCREGISTER_ALOCYEAR_TRG x

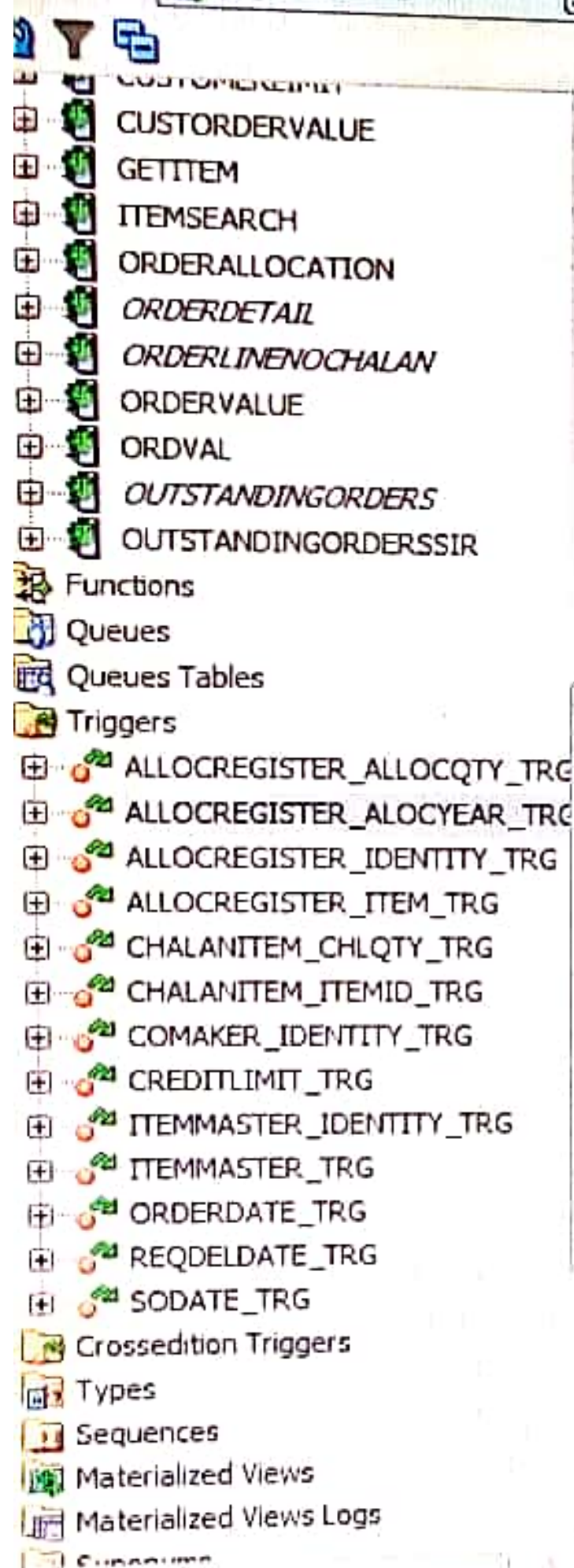
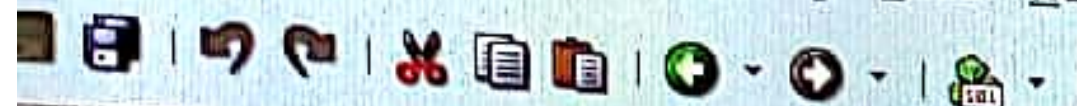
Code Grants Dependencies References Errors Details Profiles

Find

create or replace
 TRIGGER cse40."CREDITLIMIT_TRG"
 BEFORE INSERT OR UPDATE OF CREDITLIMIT ON cse40.CUSTOMER
 REFERENCING OLD AS OLDVAL NEW AS NEWVAL
 FOR EACH ROW
 DECLARE
 V_ORDER_DATE DATE;
 BEGIN
 -- REQDELDATE Must >= OrderDate + 7 days
 <<COLUMN_REQDELDATE>>
 BEGIN
 CASE
 WHEN :NEWVAL.CUSTTYPE = 'PER' THEN
 IF :NEWVAL.CREDITLIMIT <> 0 THEN
 RAISE_APPLICATION_ERROR(-20200, 'No limit permitted for Individual');
 END IF;
 WHEN :NEWVAL.CUSTTYPE = 'INS' THEN
 IF :NEWVAL.CREDITLIMIT > 5000 THEN
 RAISE_APPLICATION_ERROR(-20200, 'Max Rs.5000 permitted for Institute');
 END IF;
 WHEN :NEWVAL.CUSTTYPE = 'COM' THEN
 IF :NEWVAL.CREDITLIMIT > 100000 THEN
 RAISE_APPLICATION_ERROR(-20200, 'Max limit 100000 permitted for company');
 END IF;
 WHEN :NEWVAL.CUSTTYPE = 'SMU' THEN
 IF :NEWVAL.CREDITLIMIT > 10000 THEN
 RAISE_APPLICATION_ERROR(-20200, 'Max limit 10000 permitted for SMU');
 END IF;
 END CASE;
 EXCEPTION WHEN NO_DATA_FOUND THEN NULL;
 END COLUMN_REQDELDATE;

CUSTOMERMASTER
 CUSTORDERVALUE
 GETITEM
 ITEMSEARCH
 ORDERALLOCATION
 ORDERDETAIL
 ORDERLINENOCHALAN
 ORDERVALUE
 ORDVAL
 OUTSTANDINGORDERS
 OUTSTANDINGORDERSSIR
 Functions
 Queues
 Queues Tables
 Triggers
 ALLOCREGISTER_ALLOCQTY_TRG
 ALLOCREGISTER_ALOCYEAR_TRG
 ALLOCREGISTER_IDENTITY_TRG
 ALLOCREGISTER_ITEM_TRG
 CHALANITEM_CHLQTY_TRG
 CHALANITEM_ITEMID_TRG
 COMAKER_IDENTITY_TRG
 CREDITLIMIT_TRG
 ITEMMASTER_IDENTITY_TRG
 ITEMMASTER_TRG
 ORDERDATE_TRG
 REQDELDATE_TRG
 SODATE_TRG
 Crossedition Triggers
 Types
 Sequences
 Materialized Views
 Materialized Views Logs
 Supersums

cse40.CREDITLIMIT_TRG@cse40



```
create or replace
TRIGGER cse40."ALLOCREGISTER_ALOCYEAR_TRG"
BEFORE INSERT OR UPDATE OF ALOCYEAR ON cse40.ALLOCATIONREGISTER
FOR EACH ROW
DECLARE
    CUR_YEAR NUMBER(4,0);
BEGIN
    SELECT extract(year from sysdate) INTO CUR_YEAR FROM dual;
    IF ( :NEW.ALOCYEAR < CUR_YEAR AND :NEW.ALOCYEAR > CUR_YEAR + 1 )
        THEN RAISE_APPLICATION_ERROR(-20400, 'Allocation Year must be current year or next year ');
    END IF;
    EXCEPTION WHEN NO_DATA_FOUND THEN NULL;
END;
```


Developer

View Navigate Run Versioning Tools Help

ons x Reports x

.sql x cse40 x ORDERVALUE x ORDVAL x OUTSTANDINGORDERS x OUTSTANDINGORDERSSIR x ALLOCREGISTER_ALLOC

Code Grants Dependencies References Errors Details Profiles

Find

create or replace
TRIGGER cse40."ALLOCREGISTER_ITEM_TRG"
BEFORE INSERT OR UPDATE ON cse40.ALLOCATIONREGISTER
FOR EACH ROW
DECLARE
 v_ITEM int;
 v_ORDNO varchar2(15);
 v_ORDLINE int;

BEGIN
 -- if (updating('ORDERNO')) -- Check whether this field is part of update statement...
 SELECT :NEW.ORDERNO INTO v_ORDNO FROM DUAL ;
 --else SELECT :OLD.ORDERNO INTO v_ORDNO FROM DUAL;
 --end if;
 --if (updating('ORDERLINENO'))
 SELECT :NEW.ORDERLINENO INTO v_ORDLINE FROM DUAL;
 -- else SELECT :OLD.ORDERLINENO INTO v_ORDLINE FROM DUAL;
 --end if;
 SELECT ORDERLINEITEM.ITEMID INTO v_ITEM
 FROM ORDERLINEITEM
 WHERE ORDERLINEITEM.ORDERNO = v_ORDNO AND ORDERLINEITEM.ORDERLINENO = v_ORDLINE ;

 IF (:NEW.ITEMID <> v_ITEM)
 THEN RAISE_APPLICATION_ERROR(-20500, 'Item does not match with order line item code');
 END IF;
 EXCEPTION WHEN NO_DATA_FOUND THEN NULL;

END;

Functions
Queues
Queues Tables
Triggers
 ALLOCREGISTER_ALLOCQTY_TRG
 ALLOCREGISTER_ALOCYEAR_TRG
 ALLOCREGISTER_IDENTITY_TRG
 ALLOCREGISTER_ITEM_TRG
 CHALANITEM_CHLQTY_TRG
 CHALANITEM_ITEMID_TRG
 COMAKER_IDENTITY_TRG
 CREDITLIMIT_TRG
 ITEMMASTER_IDENTITY_TRG
 ITEMMASTER_TRG
 ORDERDATE_TRG
 REQDELDATE_TRG
 SODATE_TRG
Crossedition Triggers
Types
Sequences
Materialized Views
Materialized Views Logs

Navigate Run Versioning Tools Help

Reports x

Code Grants Dependencies References Errors Details Profiles

Find

tered)
CATIONREGISTER
LEGROUP
STER
ANHEADER
ANLINEITEM
KER
OMER
IASTER
RHEADER
RLINEITEM

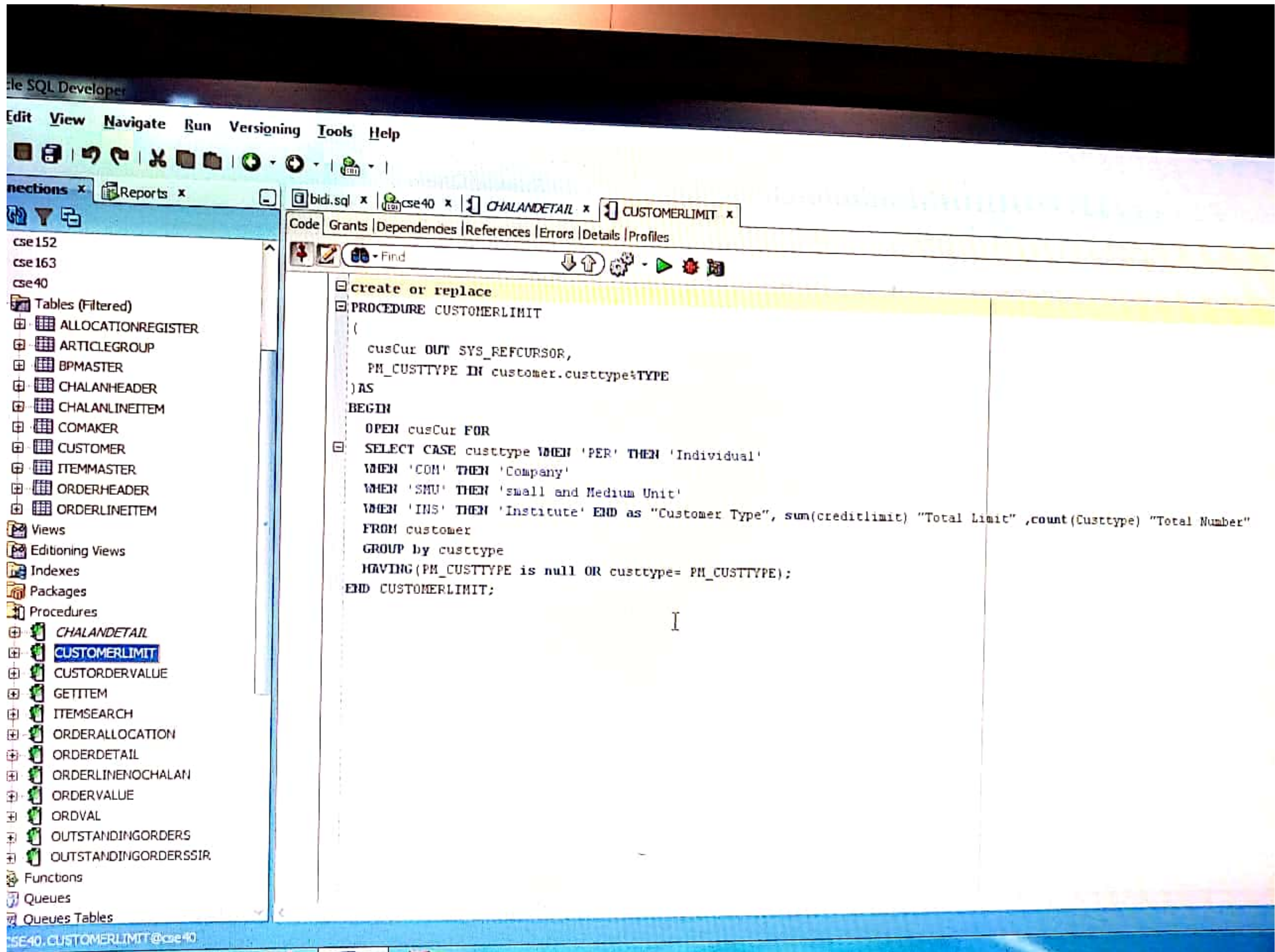
Views

DETAIL
MERLIMIT
CUSTORDERVALUE
M
SEARCH
ALLOCATION
DETAIL
LINENOCHALAN
VALUE
L
ANDINGORDERS
ANDINGORDERSSIR

les

CUSTORDERVALUE@cse40

```
create or replace
PROCEDURE CUSTORDERVALUE
(
    ordCur OUT SYS_REFCURSOR,
    PM_CUSTOMERID IN char
)AS
BEGIN
    open ordCur FOR
    SELECT e.CUSTOMERID,e.CUSTOMERNAME,MAX(g.NOOFD) "NO OF ORDER",SUM(d.ORDVAL),MAX(g.CURRENCY) CURRENCY,SUM(ORDVAL)"ORDER VALUE"
    FROM CUSTOMER e
    LEFT OUTER JOIN
    (
        SELECT COUNT(ORDERNO) "NOOFD",COUNT(CURRENCY) CURRENCY,CUSTOMERID
        FROM ORDERHEADER
        GROUP BY CUSTOMERID
    )g ON e.CUSTOMERID = g.CUSTOMERID
    LEFT OUTER JOIN
    (
        SELECT a.ORDERNO,a.CUSTOMERID,a.CURRENCY,b.ORDERQTY,(b.ORDERQTY*c.LISTPRICE)"ORDVAL"
        FROM ORDERHEADER a
        INNER JOIN ORDEPLINEITEM b ON a.ORDERNO = b.ORDERNO
        INNER JOIN ITEMMASTER c ON b.ITEMID= c.ITEMID
    )d ON e.CUSTOMERID = d.CUSTOMERID
    GROUP BY e.CUSTOMERNAME,e.CUSTOMERID
    HAVING(PM_CUSTOMERID is null OR e.CUSTOMERID=PM_CUSTOMERID)
    ORDER BY e.CUSTOMERNAME;
END CUSTORDERVALUE;
```

connections * Reports * .sql * cse40 * ALLOCREGISTER_IDENTITY_TRG * ALLOCREGISTER_ITEM_TRG * CHALANITEM_CHLQTY_TRG

Code Grants Dependencies References Errors Details Profiles

create or replace
TRIGGER cse40.ITEMMASTER_TRG
BEFORE INSERT ON cse40.ITEMMASTER
FOR EACH ROW
BEGIN
-- COLUMN SEQUENCES--
BEGIN
NULL;
END COLUMN_SEQUENCES;
END;

Functions
Queues
Queues Tables
Triggers
ALLOCREGISTER_ALLOCQTY_TRG
ALLOCREGISTER_ALOCYEAR_TRG
ALLOCREGISTER_IDENTITY_TRG
ALLOCREGISTER_ITEM_TRG
CHALANITEM_CHLQTY_TRG
CHALANITEM_ITEMID_TRG
COMAKER_IDENTITY_TRG
CREDITLIMIT_TRG
ITEMMASTER_IDENTITY_TRG
ITEMMASTER_TRG
ORDERDATE_TRG
REQDELDATE_TRG
SODATE_TRG
Crossedition Triggers
Types
Sequences
Materialized Views
Materialized Views Logs

view Navigate Run Versioning Tools Help

Icons: Save, Undo, Redo, Cut, Copy, Paste, Find, Run, Refresh, etc.

Windows: Reports x, .sql x, cse40 x, ORDERLINENOCHALAN x, ORDERVALUE x, ORDVAL x, OUTSTANDINGORDERS x, OUTSTANDING

Code Grants Dependencies References Errors Details Profiles

Find

create or replace

```
TRIGGER cse40."ALLOCREGISTER_IDENTITY_TRG"
BEFORE INSERT ON cse40.ALLOCATIONREGISTER
FOR EACH ROW
BEGIN
  <<COLUMN_SEQUENCES>>
  BEGIN
    IF INSERTING AND :NEW.ALLOCNO IS NULL THEN
      SELECT ALLOCATIONREGISTER_SEQ.NEXTVAL INTO :NEW.ALLOCNO FROM SYS.DUAL;
    END IF;
  END COLUMN_SEQUENCES;
END;
```

Left Panel (Database Objects):

- FUNCTIONS
- Queues
- Queues Tables
- Triggers
 - ALLOCREGISTER_ALLOCQTY_TRG
 - ALLOCREGISTER_ALOCYEAR_TRG
 - ALLOCREGISTER_IDENTITY_TRG
 - ALLOCREGISTER_ITEM_TRG
 - CHALANITEM_CHLQTY_TRG
 - CHALANITEM_ITEMID_TRG
 - COMAKER_IDENTITY_TRG
 - CREDITLIMIT_TRG
 - ITEMMASTER_IDENTITY_TRG
 - ITEMMASTER_TRG
 - ORDERDATE_TRG
 - REQDELDATE_TRG
 - SODATE_TRG
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized Views Logs
- Suppliers

Reports x

cse40 x CUSTOMERLIMIT x CUSTORDERVALUE x GETITEM x ITEMSEARCH x ORDERALLOCATION x

Code Grants Dependencies References Errors Details Profiles

Find

les (Filtered)

- ALLOCATIONREGISTER
- ARTICLEGROUP
- BPMaster
- CHALANHEADER
- CHALANLINEITEM
- COMAKER
- CUSTOMER
- ITEMMASTER
- ORDERHEADER
- ORDERLINEITEM

ws

tioning Views

lexes

ckages

cedures

- CHALANDETAIL
- CUSTOMERLIMIT
- CUSTORDERVALUE
- GETITEM
- ITEMSEARCH
- ORDERALLOCATION
- ORDERDETAIL
- ORDERLINENOCHALAN
- ORDERVALUE
- ORDVAL
- OUTSTANDINGORDERS
- OUTSTANDINGORDERSSIR

inctions

roups

```
create or replace
PROCEDURE ORDERVALUE
(
    ordCur OUT SYS_REFCURSOR
) AS
BEGIN
    OPEN ordCur FOR
    SELECT SUM(a.ORDERQTY), MAX(c.CUSTOMERNAME), SUM(a.ORDERQTY*b.LISTPRICE) "ORDER VALUE"
    FROM ORDERLINEITEM a
    INNER JOIN ITEMMASTER b ON a.ITEMID=b.ITEMID
    INNER JOIN ORDERHEADER d ON a.ORDERNO=d.ORDERNO
    INNER JOIN CUSTOMER c ON d.CUSTOMERID = c.CUSTOMERID
    GROUP BY c.CUSTOMERID;
END ORDERVALUE;
```




Actions x Reports x

- CUSTOMERITEM
- CUSTORDERVALUE
- GETITEM
- ITEMSEARCH
- ORDERALLOCATION
- ORDERDETAIL
- ORDERLINENOCHALAN
- ORDERVALUE
- ORDVAL
- OUTSTANDINGORDERS
- OUTSTANDINGORDERSSIR

Functions

Queues

Queues Tables

Triggers

- ALLOCREGISTER_ALLOCQTY_TRG
- ALLOCREGISTER_ALOCYEAR_TRG
- ALLOCREGISTER_IDENTITY_TRG
- ALLOCREGISTER_ITEM_TRG
- CHALANITEM_CHLQTY_TRG
- CHALANITEM_ITEMID_TRG
- COMAKER_IDENTITY_TRG
- CREDITLIMIT_TRG
- ITEMMASTER_IDENTITY_TRG
- ITEMMASTER_TRG
- ORDERDATE_TRG**
- REQDELDATE_TRG
- SODATE_TRG

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized Views Logs

Code Grants Dependencies References Errors Details Profiles

Find

create or replace

TRIGGER cse40."ORDERDATE_TRG"

BEFORE INSERT OR UPDATE OF ORDERDATE ON cse40.ORDERHEADER

FOR EACH ROW

-- A check constraint, unfortunately, cannot reference a function like SYSDATE. You would need
-- this values when DML occurs

BEGIN

IF(:new.ORDERDATE > ADD_MONTHS(SYSDATE, -1))

THEN

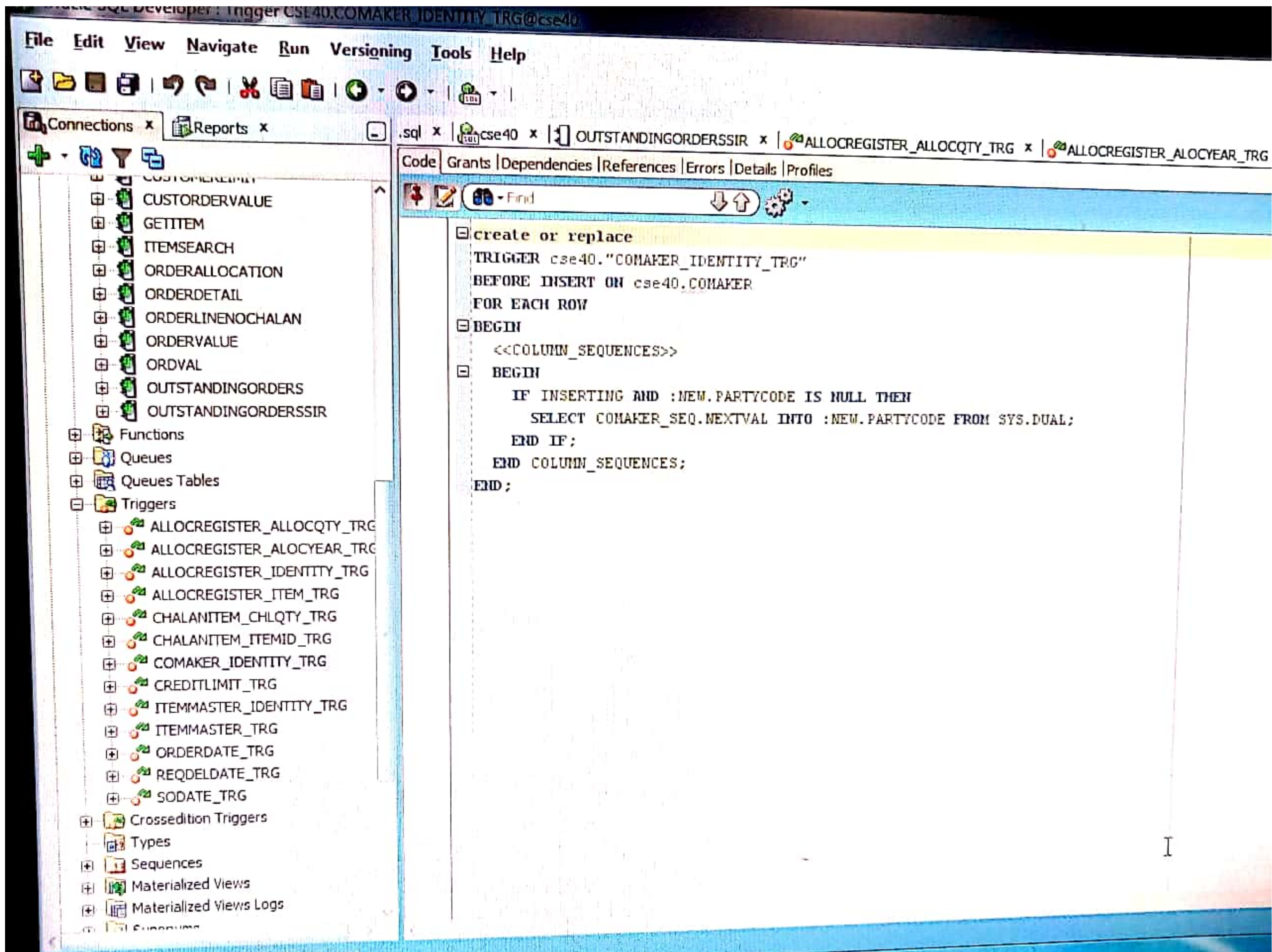
RAISE_APPLICATION_ERROR(-20001,

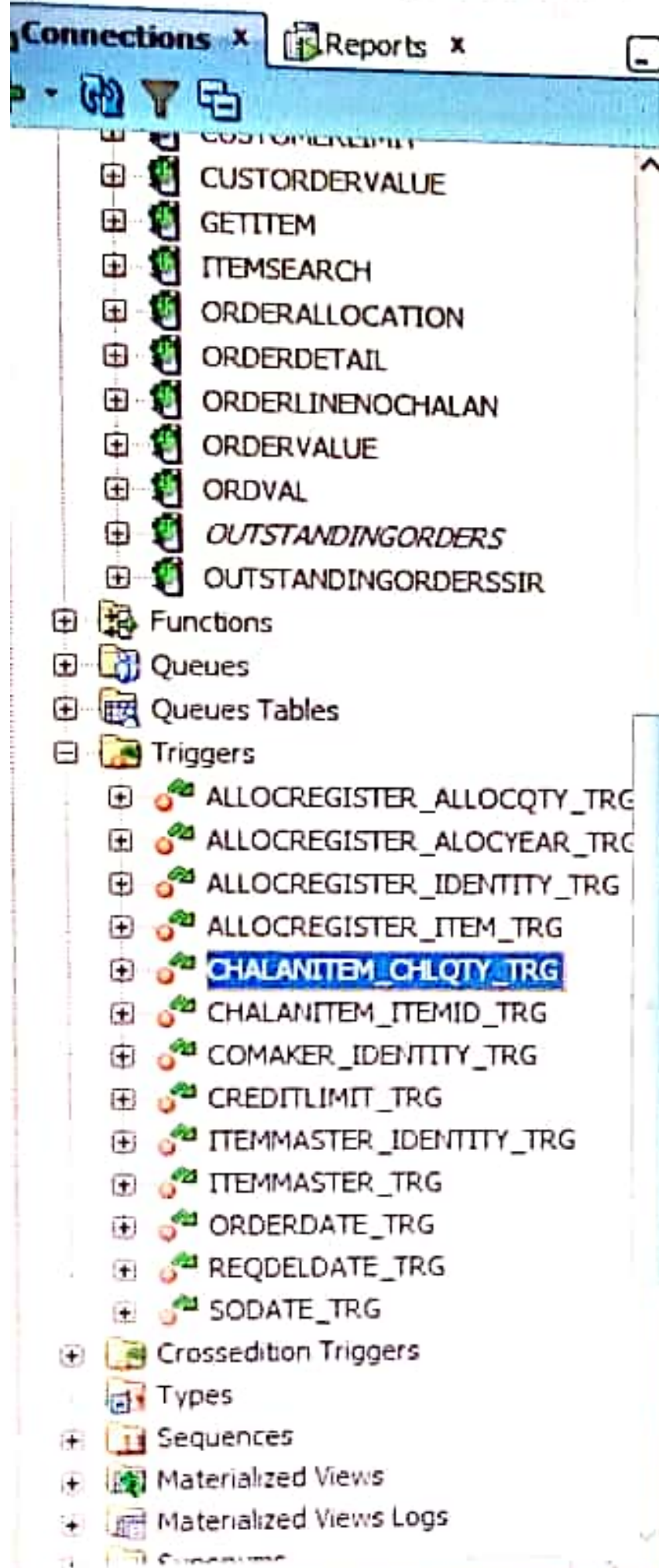
'Order date must be less than the current date ');

END IF;

END;

I





Code Grants Dependencies References Errors Details Profiles

Find

```
create or replace
TRIGGER cse40."CHALANITEM_CHLQTY_TRG"
BEFORE INSERT OR UPDATE OF CHLQTY ON cse40.CHALANLINEITEM
FOR EACH ROW
DECLARE
    V_ORDERQTY int;
BEGIN
    SELECT ORDERLINEITEM.ORDERQTY INTO V_ORDERQTY
    FROM ORDERLINEITEM
    WHERE ORDERLINEITEM.ORDERNO = :NEW.ORDERNO AND ORDERLINEITEM.ORDERLINENO = :NEW.ORDERLINENO ;

    IF ( :NEW.CHLQTY > V_ORDERQTY )
        THEN RAISE_APPLICATION_ERROR(-20600, 'Must be <= order quantity ');
    END IF;
    EXCEPTION WHEN NO_DATA_FOUND THEN NULL;

END;
```


Functions x Reports x

.sql x cse40 x ALLOCREGISTER_IDENTITY_TRG x ALLOCREGISTER_ITEM_TRG x CHALANITEM_CHLQTY_TRG x

Code Grants Dependencies References Errors Details Profiles

Find

create or replace
TRIGGER cse40."ITEMMASTER_TRG"
BEFORE INSERT ON cse40.ITEMMASTER
FOR EACH ROW
BEGIN
 <<COLUMN_SEQUENCES>>
 BEGIN
 NULL;
 END COLUMN_SEQUENCES;
END;

Functions
Queues
Queues Tables
Triggers
+ ALLOCREGISTER_ALLOCQTY_TRG
+ ALLOCREGISTER_ALOCYEAR_TRG
+ ALLOCREGISTER_IDENTITY_TRG
+ ALLOCREGISTER_ITEM_TRG
+ CHALANITEM_CHLQTY_TRG
+ CHALANITEM_ITEMID_TRG
+ COMAKER_IDENTITY_TRG
+ CREDITLIMIT_TRG
+ ITEMMASTER_IDENTITY_TRG
+ **ITEMMASTER_TRG**
+ ORDERDATE_TRG
+ REQDELDATE_TRG
+ SODATE_TRG
Crossedition Triggers
Types
Sequences
Materialized Views
Materialized Views Logs
Supervisors



- CUSTOMER
- CUSTOMERVALUE
- GETITEM
- ITEMSEARCH
- ORDERALLOCATION
- ORDERDETAIL
- ORDERLINENOCHALAN
- ORDERVALUE
- ORDVAL
- OUTSTANDINGORDERS
- OUTSTANDINGORDERSIR

Functions

Queues

Queues Tables

Triggers

- ALLOCREGISTER_ALLOCQTY_TRG
- ALLOCREGISTER_ALOCYEAR_TRG
- ALLOCREGISTER_IDENTITY_TRG
- ALLOCREGISTER_ITEM_TRG
- CHALANITEM_CHLQTY_TRG
- CHALANITEM_ITEMID_TRG
- COMAKER_IDENTITY_TRG
- CREDITLIMIT_TRG
- ITEMMASTER_IDENTITY_TRG
- ITEMMASTER_TRG
- ORDERDATE_TRG
- REQDELDATE_TRG
- SODATE_TRG

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized Views Logs

Functions

.sql x | cse40 x | ORDERALLOCATION x | ORDERDETAIL x | ORDERLINENOCHALAN x | ORDERVALUE x | ORDVAL x | OUTSTANDINGORDERS x |

Code Grants Dependencies References Errors Details Profiles

Find



create or replace

TRIGGER cse40."ALLOCREGISTER_ALLOCQTY_TRG"

BEFORE INSERT OR UPDATE OF ALLOCQTY ON cse40.ALLOCATIONREGISTER

FOR EACH ROW

DECLARE

v_ORDQTY int;

BEGIN

IF UPDATING THEN

SELECT ORDERLINEITEM.ORDERQTY INTO v_ORDQTY

FROM ORDERLINEITEM

WHERE ORDERLINEITEM.ORDERNO = :OLD.ORDERNO AND ORDERLINEITEM.ORDERLINENO = :OLD.ORDERLINENO ;

END IF;

IF INSERTING THEN

SELECT ORDERLINEITEM.ORDERQTY INTO v_ORDQTY

FROM ORDERLINEITEM

WHERE ORDERLINEITEM.ORDERNO = :NEW.ORDERNO AND ORDERLINEITEM.ORDERLINENO = :NEW.ORDERLINENO ;

END IF;

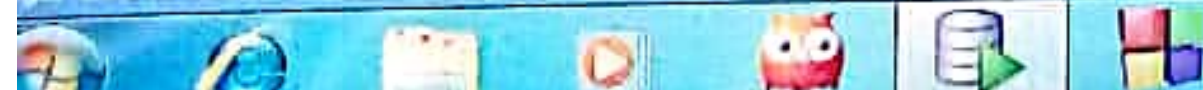
IF (:NEW.ALLOCQTY > v_ORDQTY OR :NEW.ALLOCQTY <= 0)

THEN RAISE_APPLICATION_ERROR(-20400, 'Allocation Qty must be > 0 and less than order qty');

END IF;

EXCEPTION WHEN NO_DATA_FOUND THEN NULL;

END;



SQL Developer

File View Navigate Run Versioning Tools Help

Actions x Reports x

se152
se163
se40

Tables (Filtered)

- ALLOCATIONREGISTER
- ARTICLEGROUP
- BPMASER
- CHALANHEADER
- CHALANLINEITEM
- COMAKER
- CUSTOMER
- ITEMMASTER
- ORDERHEADER
- ORDERLINEITEM

Views

Editing Views

Indexes

Packages

Procedures

- CHALANDETAIL
- CUSTOMERLIMIT
- CUSTORDERVALUE
- GETITEM
- ITEMSEARCH
- ORDERALLOCATION
- ORDERDETAIL
- ORDERLINENOCHALAN
- ORDERVALUE
- ORDVAL**
- OUTSTANDINGORDERS
- OUTSTANDINGORDERSSIR

Functions

Queues

Queues Tables

CSE40.ORDVAL@cse40

Code Grants Dependencies References Errors Details Profiles

Find

```
create or replace
PROCEDURE ORDVAL
(
    ordCur OUT SYS_REFCURSOR
)AS
BEGIN
    OPEN ordCur FOR
    SELECT a.ORDERNO,a.ORDERDATE,d.CUSTOMERNAME,a.ORDERBP,a.CURRENCY,e.ORDQTY "ORDER QTY", e.ORDVALUE "order Value"
    FROM
    (
        SELECT b.ORDERNO,SUM(b.ORDERQTY) "ORDQTY", SUM(b.ORDERQTY *c.LISTPRICE) "ORDVALUE"
        FROM ORDERLINEITEM b
        INNER JOIN ITEMMASTER c ON b.ITEMID=c.ITEMID
        GROUP BY b.ORDERNO
    )e
    INNER JOIN ORDERHEADER a ON e.ORDERNO=a.ORDERNO
    INNER JOIN CUSTOMER d ON a.CUSTOMERID=d.CUSTOMERID
    ORDER BY a.ORDERNO;
END ORDVAL;
```




- Actions x
- Reports x
- Functions
- Queues
- Queues Tables
- Triggers
 - ALLOCREGISTER_ALLOCQTY_TRG
 - ALLOCREGISTER_ALOCYEAR_TRG
 - ALLOCREGISTER_IDENTITY_TRG
 - ALLOCREGISTER_ITEM_TRG
 - CHALANITEM_CHLQTY_TRG
 - CHALANITEM_ITEMID_TRG
 - COMAKER_IDENTITY_TRG
 - CREDITLIMIT_TRG
 - ITEMMASTER_IDENTITY_TRG
 - ITEMMASTER_TRG
 - ORDERDATE_TRG
 - REQDELDATE_TRG
 - SODATE_TRG**
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized Views Logs
- Summary

Code Grants Dependencies References Errors Details Profiles

Find

```

create or replace
TRIGGER cse40."SODATE_TRG"
BEFORE INSERT OR UPDATE OF SODATE ON cse40.ORDERLINEITEM
FOR EACH ROW
DECLARE
    V_ORDER_DATE DATE;
BEGIN
    -- SODATE if not null <= OrderDate
    <<COLUMN_SODATE>>
BEGIN
    SELECT ORDERHEADER.ORDERDATE INTO V_ORDER_DATE
    FROM ORDERHEADER
    WHERE ORDERHEADER.ORDERNO = :NEW.ORDERNO;

    IF ( :NEW.SODATE IS NOT NULL ) AND :NEW.SODATE > V_ORDER_DATE
        THEN RAISE_APPLICATION_ERROR(-20300, 'SODATE must be <= ORDERDATE ');
    END IF;
    EXCEPTION WHEN NO_DATA_FOUND THEN NULL;
END COLUMN_SODATEDATE;
END;
    
```


edit View Navigate Run Versioning Tools Help

Sections x Reports x

cse 152
cse 163
cse 40

Tables (Filtered)

- ALLOCATIONREGISTER
- ARTICLEGROUP
- BPMaster
- CHALANHEADER
- CHALANLINEITEM
- COMAKER
- CUSTOMER
- ITEMMASTER
- ORDERHEADER
- ORDERLINEITEM

Views

Editing Views

Indexes

Packages

Procedures

- CHALANDETAIL
- CUSTOMERLIMIT
- CUSTORDERVALUE
- GETITEM
- ITEMSEARCH
- ORDERALLOCATION
- ORDERDETAIL
- ORDERLINENOCHALAN
- ORDERVALUE
- ORDVAL
- OUTSTANDINGORDERS
- OUTSTANDINGORDERSSIR

Functions

Queues

Queues Tables

.sql x cse40 x GETITEM x ITEMSEARCH x ORDERALLOCATION x ORDERDETAIL x ORDERLINENOCHALAN x OR

Code Grants Dependencies References Errors Details Profiles

Find

```
create or replace
PROCEDURE OUTSTANDINGORDERS
(
    PM_orderno in orderlineitem.orderno%type,
    ordcur out sys_refcursor
)AS
BEGIN
    open ordcur FOR
    select max(a.orderno),max(c.customerid),max(a.orderqty),sum(e.itemid),
    max(e.itemdescription),max(c.customername),sum(a.orderqty-b.chlqty) "outstanding order"
    from orderlineitem a
    inner join chalanlineitem b on a.orderno=b.orderno
    inner join orderheader d on a.orderno = d.orderno
    inner join customer c on d.customerid=c.customerid
    inner join itemmaster e on a.itemid=e.itemid
    group by a.orderno;

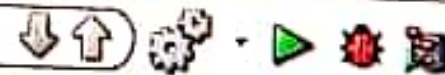
END OUTSTANDINGORDERS;
```

Line 10

re CSE40.OUTSTANDINGORDERS@cse40



- cse152
- cse163
- cse40
 - Tables (Filtered)
 - ALLOCATIONREGISTER
 - ARTICLEGROUP
 - BPMASER
 - CHALANHEADER
 - CHALANLINEITEM
 - COMAKER
 - CUSTOMER
 - ITEMMASTER
 - ORDERHEADER
 - ORDERLINEITEM
 - Views
 - Editing Views
 - Indexes
 - Packages
 - Procedures
 - CHALANDETAIL
 - CUSTOMERLIMIT
 - CUSTORDERVALUE
 - GETITEM
 - ITEMSEARCH
 - ORDERALLOCATION
 - ORDERDETAIL
 - ORDERLINENOCHALAN
 - ORDERVALUE
 - ORDVAL
 - OUTSTANDINGORDERS
 - OUTSTANDINGORDERSSIR
 - Functions
 - Queues
 - Queues Tables



create or replace

PROCEDURE GETITEM

```
(  
    itmCur OUT SYS_REFCURSOR,  
    PM_ITEMCAT IN itemmaster.itemcode%TYPE,  
    PM_ITEMTYPE IN VARCHAR2  
) AS  
BEGIN  
    OPEN itmCur FOR  
    SELECT a.Itemid,a.Itemcode,a.Articlegr,a.Itemcat,a.Itentype,a.Itendescription,b.ArticlegrDescription  
    FROM ITEMMASTER a  
    INNER JOIN ARTICLEGROUP b ON a.Articlegr = b.Articlegr  
    WHERE (PM_ITEMCAT is null OR a.Itemcat=PM_ITEMCAT)  
    AND (PM_ITEMTYPE is null OR a.Itentype = PM_ITEMTYPE)  
    ORDER BY a.Itemcode;  
END GETITEM;
```


ew **Navigate** **Run** **Versioning** **Tools** **Help**

Reports x | biddi.sql x | cse40 x | CHALANDETAIL x | CUSTOMERLIMIT x | CUSTORDERVALUE x | GETITEM x | ITEMSEARCH x | ORD

Code Grants Dependences References Errors Details Profiles

Find

es (Filtered)
ALLOCATIONREGISTER
ARTICLEGROUP
BPMASER
CHALANHEADER
CHALANLINEITEM
COMAKER
CUSTOMER
ITEMMASTER
ORDERHEADER
ORDERLINEITEM
is
oning Views
xes
ages
edures
CHALANDETAIL
CUSTOMERLIMIT
CUSTORDERVALUE
GETITEM
ITEMSEARCH
ORDERALLOCATION
ORDERDETAIL
ORDERLINENOCHALAN
ORDERVALUE
ORDVAL
OUTSTANDINGORDERS
OUTSTANDINGORDERSSIR
ctions
ues
nes Tables

```
create or replace
PROCEDURE ORDERDETAIL
(
    ordCur OUT SYS_REFCURSOR
)AS
BEGIN
    OPEN ordCur FOR
    SELECT a.ORDERNO, a.ORDERDATE, b.ORDERLINENO, a.SALESOFFICER, a.CUSTOMERID,
    d.CUSTOMERNAME, a.ORDERTYPE, a.ORDERBP, a.CURRENCY, b.ITEMID, c.ITEMDESCRIPTION,
    b.SONO, b.SODATE, b.SOQTY, b.ORDERQTY, (b.ORDERQTY*c.LISTPRICE) "ORDER VALUE > 600000"
    FROM ORDERHEADER a
    INNER JOIN ORDERLINEITEM b ON a.ORDERNO = b.ORDERNO
    INNER JOIN ITEMMASTER c ON b.ITEMID = c.ITEMID
    INNER JOIN CUSTOMER d ON a.CUSTOMERID = d.CUSTOMERID
    ORDER BY a.ORDERNO, b.ORDERLINENO;
END ORDERDETAIL;
```


SQL Developer

View Navigate Run Versioning Tools Help

Actions x Reports x

.sql x cse40 x CHALANITEM_CHLQTY_TRG x CHALANITEM_ITEMID_TRG x COMAKER_IDENTITY_TRG x COMAKER_IDENTITY

Code Grants Dependencies References Errors Details Profiles

Find

create or replace
TRIGGER cse40."REQDELDATE_TRG"
BEFORE INSERT OR UPDATE OF REQDELDATE ON cse40.ORDERLINEITEM
FOR EACH ROW
DECLARE
 V_ORDER_DATE DATE;
BEGIN
 -- REQDELDATE Must >= OrderDate + 7 days
 <<COLUMN_REQDELDATE>>
BEGIN
 SELECT ORDERHEADER.ORDERDATE INTO V_ORDER_DATE
 FROM ORDERHEADER
 WHERE ORDERHEADER.ORDERNO = :NEW.ORDERNO;

 IF :NEW.REQDELDATE <= V_ORDER_DATE + 7
 THEN RAISE_APPLICATION_ERROR(-20100, 'REQDELDATE must be >= ORDERDATE + 7');
 END IF;
 EXCEPTION WHEN NO_DATA_FOUND THEN NULL;
END COLUMN_REQDELDATE;
END;

CUSTOMERITEM
CUSTORDERVALUE
GETITEM
ITEMSEARCH
ORDERALLOCATION
ORDERDETAIL
ORDERLINENOCHALAN
ORDERVALUE
ORDVAL
OUTSTANDINGORDERS
OUTSTANDINGORDERSSIR
Functions
Queues
Queues Tables
Triggers
ALLOCREGISTER_ALLOCQTY_TRG
ALLOCREGISTER_ALOCYEAR_TRG
ALLOCREGISTER_IDENTITY_TRG
ALLOCREGISTER_ITEM_TRG
CHALANITEM_CHLQTY_TRG
CHALANITEM_ITEMID_TRG
COMAKER_IDENTITY_TRG
CREDITLIMIT_TRG
ITEMMASTER_IDENTITY_TRG
ITEMMASTER_TRG
ORDERDATE_TRG
REQDELDATE_TRG
SODATE_TRG
Crossedition Triggers
Types
Sequences
Materialized Views
Materialized Views Logs
Supervisors

View Navigate Run Versioning Tools Help

Actions x Reports x

.sql x cse40 x ALLOCREGISTER_ALOCYEAR_TRG x ALLOCREGISTER_ALOCYEAR_TRG x ALLOCREGISTER_IDENTITY_TRG

Code Grants Dependencies References Errors Details Profiles

Find

create or replace
TRIGGER cse40."ITEMMASTER_IDENTITY_TRG"
BEFORE INSERT ON cse40.ITEMMASTER
FOR EACH ROW
BEGIN
 <<COLUMN_SEQUENCES>>
 BEGIN
 IF INSERTING AND :NEW.ITEMID IS NULL THEN
 SELECT ITEMMASTER_SEQ.NEXTVAL INTO :NEW.ITEMID FROM SYS.DUAL;
 END IF;
 END COLUMN_SEQUENCES;
END;

CUSTOMERLIMIT
CUSTORDERVALUE
GETITEM
ITEMSEARCH
ORDERALLOCATION
ORDERDETAIL
ORDERLINENOCHALAN
ORDERVALUE
ORDVAL
OUTSTANDINGORDERS
OUTSTANDINGORDERSSIR
Functions
Queues
Queues Tables
Triggers
ALLOCREGISTER_ALLOCQTY_TRG
ALLOCREGISTER_ALOCYEAR_TRG
ALLOCREGISTER_IDENTITY_TRG
ALLOCREGISTER_ITEM_TRG
CHALANITEM_CHLQTY_TRG
CHALANITEM_ITEMID_TRG
COMAKER_IDENTITY_TRG
CREDITLIMIT_TRG
ITEMMASTER_IDENTITY_TRG
ITEMMASTER_TRG
ORDERDATE_TRG
REQDELDATE_TRG
SODATE_TRG
Crossedition Triggers
Types
Sequences
Materialized Views
Materialized Views Logs
Supervisors

Developer

View Navigate Run Versioning Tools Help

Reports x

Code Grants Dependencies References Errors Details Profiles

Find

create or replace
PROCEDURE ITEMSEARCH

```
(  
    ITMCUR OUT SYS_REFCURSOR,  
    P_ITEMCAT IN ITEMMASTER.ITEMCAT%TYPE,  
    P_ITEMTYPE IN VARCHAR2,  
    P_ITEMDES IN ITEMMASTER.ITEMDESCRIPTION%TYPE  
)AS  
BEGIN  
    OPEN ITMCUR FOR  
    SELECT a.Itemid, a.Itemcode, a.Articlegr, a.Itemcat, a.Itemtype, a.Itemdescription, b.ArticlegrDescription  
    FROM ITEMMASTER a  
    INNER JOIN ARTICLEGROUP b ON a.Articlegr = b.Articlegr  
    WHERE (P_ITEMCAT is null OR a.Itemcat=P_ITEMCAT)  
    AND (P_ITEMTYPE is null OR a.ITEMTYPE = P_ITEMTYPE)  
    AND (P_ITEMDES is null OR a.Itemdescription LIKE concat(P_ITEMDES,'%'))  
    ORDER BY a.Itemdescription;  
END ITEMSEARCH;
```

Items (Filtered)

- ALLOCATIONREGISTER
- ARTICLEGROUP
- BPMMASTER
- CHALANHEADER
- CHALANLINEITEM
- COMAKER
- CUSTOMER
- ITEMMASTER
- ORDERHEADER
- ORDERLINEITEM

Views

Filtering Views

Indexes

Packages

Procedures

- CHALANDETAIL
- CUSTOMERLIMIT
- CUSTORDERVALUE
- GETITEM
- ITEMSEARCH**
- ORDERALLOCATION
- ORDERDETAIL
- ORDERLINENOCHALAN
- ORDERVALUE
- ORDVAL
- OUTSTANDINGORDERS
- OUTSTANDINGORDERSSIR

Functions

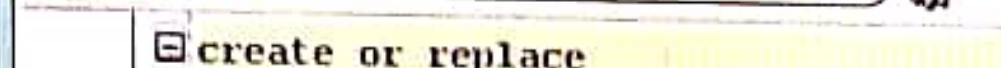
Views

Views Tables

ITEMSEARCH@cse40



- CUSTOMERITEM
 - CUSTORDERVALUE
 - GETITEM
 - ITEMSEARCH
 - ORDERALLOCATION
 - ORDERDETAIL
 - ORDERLINENOCHALAN
 - ORDERVALUE
 - ORDVAL
 - OUTSTANDINGORDERS
 - OUTSTANDINGORDERSSIR
- Functions
- Queues
- Queues Tables
- Triggers
 - ALLOCREGISTER_ALLOCQTY_TRG
 - ALLOCREGISTER_ALOCYEAR_TRG
 - ALLOCREGISTER_IDENTITY_TRG
 - ALLOCREGISTER_ITEM_TRG
 - CHALANITEM_CHLQTY_TRG
 - CHALANITEM_ITEMID_TRG**
 - COMAKER_IDENTITY_TRG
 - CREDITLIMIT_TRG
 - ITEMMASTER_IDENTITY_TRG
 - ITEMMASTER_TRG
 - ORDERDATE_TRG
 - REQDELDATE_TRG
 - SODATE_TRG
- Crossedition Triggers
- Types
- Sequences
- Materialized Views
- Materialized Views Logs
- Supervisors



```

create or replace
TRIGGER cse40."CHALANITEM_ITEMID_TRG"
BEFORE INSERT OR UPDATE OF ITEMID ON cse40.CHALANLINEITEM
FOR EACH ROW
DECLARE
    V_ITEMID int;
BEGIN
    SELECT ORDERLINEITEM.ITEMID INTO V_ITEMID
    FROM ORDERLINEITEM
    WHERE ORDERLINEITEM.ORDERNO = :NEW.ORDERNO AND ORDERLINEITEM.ORDERLINENO = :NEW.ORDERLINENO ;

    IF ( :NEW.ITEMID != V_ITEMID )
    THEN RAISE_APPLICATION_ERROR(-20500, 'Must be same item id of order line item ');
    END IF;
    EXCEPTION WHEN NO_DATA_FOUND THEN NULL;

END;
    
```


Connections x Reports x

cse152
cse163
cse40

Tables (Filtered)

- ALLOCATIONREGISTER
- ARTICLEGROUP
- BPMMASTER
- CHALANHEADER
- CHALANLINEITEM
- COMAKER
- CUSTOMER
- ITEMMASTER
- ORDERHEADER
- ORDERLINEITEM

Views

Editing Views

Indexes

Packages

Procedures

- CHALANDETAIL
- CUSTOMERLIMIT
- CUSTORDERVALUE
- GETITEM
- ITEMSEARCH
- ORDERALLOCATION
- ORDERDETAIL
- ORDERLINENOCHALAN
- ORDERVALUE
- ORDVAL
- OUTSTANDINGORDERS
- OUTSTANDINGORDERSSIR

Functions

Queues

Queues Tables

Code Grants Dependencies References Errors Details Profiles

Find

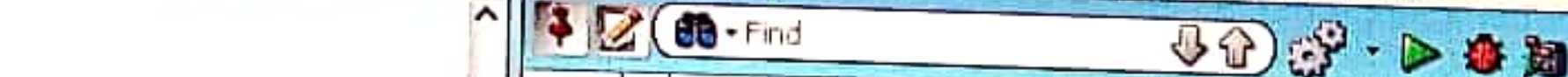
```
create or replace
PROCEDURE ORDERLINENOCHALAN
(
    ORDCUR OUT SYS_REFCURSOR,
    PM_ORDERNO IN INT,
    PM_PARTYID IN CHAR
)AS
BEGIN
    OPEN ORDCUR FOR
    SELECT a.ORDERNO,a.ORDERLINENO,b.CHALANNO,d.CHALANDATE,b.CHALANLINEITEM,c.ITEMCODE,
    b.ITEMDESCRIPTION,e.PARTYID,e.COMAKERNAME,
    b.CHLOQTY,(b.CHLOQTY*c.LISTPRICE) "CHALAN VALUE"
    FROM ORDERLINEITEM a
    LEFT OUTER JOIN CHALANLINEITEM b ON a.ORDERNO = b.ORDERNO AND
    a.ORDERLINENO = b.ORDERLINENO
    LEFT OUTER JOIN ITEMMASTER c ON b.ITEMID=c.ITEMID
    LEFT OUTER JOIN CHALANHEADER d ON b.CHALANNO = d.CHALANNO
    LEFT OUTER JOIN COMAKER e ON d.PARTYCODE = e.PARTYCODE
    WHERE (PM_ORDERNO IS NULL OR b.ORDERNO = PM_ORDERNO)AND
    (PM_PARTYID IS NULL OR e.PARTYID = PM_PARTYID)
    ORDER BY a.ORDERNO,a.ORDERLINENO,b.CHALANNO;
END ORDERLINENOCHALAN;
```

Procedure CSE40.ORDERLINENOCHALAN@cse40



Reports x bidi.sql x cse40 x CHALANDETAIL x

Code Grants Dependencies References Errors Details Profiles



es (Filtered)

ALLOCATIONREGISTER

ARTICLEGROUP

3PMMASTER

CHALANHEADER

CHALANLINEITEM

COMAKER

CUSTOMER

ITEMMASTER

ORDERHEADER

ORDERLINEITEM

;

ining Views

res

ages

cedures

CHALANDETAIL

CUSTOMERLIMIT

CUSTOMERVALUE

ETITEM

ITEMSEARCH

ORDERALLOCATION

ORDERDETAIL

ORDERLINENOCHALAN

ORDERVALUE

RDVAL

UTSTANDINGORDERS

UTSTANDINGORDERSSIR

ons

es

es Tables

CHALANDETAIL @cse40

```

create or replace
PROCEDURE CHALANDETAIL
(
    CHLNCUR OUT SYS_REFCURSOR,
    PM_ORDERNO IN VARCHAR2,
    PM_PARTYID IN VARCHAR2
) AS
BEGIN
    OPEN CHLNCUR FOR
    SELECT a.CHALANNO,a.CHALAMDATE, b.CHALANLINEITEM,b.ORDERNO,b.ORDERLINENO,d.PARTYID,d.COMAKERNAME,c.ITEMCODE,
    c.ITEMDESCRIPTION,b.CHLQTY,(b.CHLQTY*c.LISTPRICE) "CHALAN VALUE"
    FROM CHALANHEADER a
    INNER JOIN CHALANLINEITEM b ON a.CHALANNO = b.CHALANNO
    INNER JOIN ITEMMASTER c ON b.ITEMID = c.ITEMID
    INNER JOIN COMAKER d ON a.PARTYCODE = d.PARTYCODE
    WHERE (PM_ORDERNO is null OR b.ORDERNO = PM_ORDERNO)
    AND (PM_PARTYID is null OR d.PARTYID = PM_PARTYID)
    ORDER BY a.CHALANNO, b.CHALANLINEITEM;
END CHALANDETAIL;
    
```


Connections * Reports *

Code Grants Dependencies References Errors Details Profiles

create or replace

```
TRIGGER cse40.ITEMMASTER_TPG
BEFORE INSERT ON cse40.ITEMMASTER
FOR EACH ROW
BEGIN
    << COLUMN_SEQUENCES >>
    BEGIN
        NULL;
    END COLUMN_SEQUENCES;
END;
```

FUNCTIONS

QUEUES

QUEUE TABLES

TRIGGERS

- ALLOCREGISTER_ALLOCQTY_TRG
- ALLOCREGISTER_ALLOCYEAR_TRG
- ALLOCREGISTER_IDENTITY_TRG
- ALLOCREGISTER_ITEM_TRG
- CHALANITEM_CHLQTY_TRG
- CHALANITEM_ITEMID_TRG
- COMAKER_IDENTITY_TRG
- CREDITLIMIT_TRG
- ITEMMASTER_IDENTITY_TRG
- ITEMMASTER_TRG**
- ORDERDATE_TRG
- REQDEDATE_TRG
- SODATE_TRG

Crossedition Triggers

Types

Sequences

Materialized Views

Materialized views Logs