

## TAAS - Feature #147

### Customer Overdue Report

02/02/2023 03:21 PM - Anil KV

<b>Status:</b>	Reopen Bugs Coding Done	<b>Start date:</b>	02/07/2023
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Augustin Jose	<b>% Done:</b>	70%
<b>Category:</b>		<b>Estimated time:</b>	0:00 hour
<b>Target version:</b>	22.12	<b>Spent time:</b>	0:00 hour
<b>Owner(Agency):</b>	Travvise	<b>Tested By:</b>	unnikannan S
<b>Time Taken(HH):</b>		<b>Code Reviewed By:</b>	
<b>Module:</b>	Customer Overdue Report		
<b>Description</b>			
Coding Customer Overdue Report			

### History

#### #1 - 02/02/2023 03:22 PM - Anil KV

- Assignee set to Junaid M

#### #2 - 05/24/2023 06:41 PM - Anil KV

##### GUI Changes:-

- 1) Currency(Check ledger stmt) - Remove Base2(\*\*) item, Default Empty
- 2) Debit and Credit Checkbox set after \_ Consider Matching Done After To Date\_ default Tick all(3) check boxes
- 3) Add \*Consider Not Approved Documents Checkbox - Default Untick
- 4) Payment Terms row also include more(Expand) div
- 5) Payment Terms Select box need a checkbox before, Title-Tick ON then allow to Edit, Otherwise take each customers Payment Terms
- 6) All customer min ledgers load and Tick on all
- 7) ON and Disable Debit Checkbox
- 8) List #  
Code  
Name  
Currency  
Current  
-Net Balance  
-Outstanding Dr  
-Outstanding Cr  
-Net Outstanding  
As On Date  
-Net Balance  
-Outstanding Dr  
-Outstanding Cr  
-Net Outstanding  
Payment Terms  
Outstanding (Note: Consider **Consider Matching Done After To Date** input)  
Due Days  
Due Amount

#### #3 - 05/25/2023 02:22 PM - Anil KV

##### Points

- 1) Branch and Depart select all then skip where conditions

##### Currency = Base1(...) Selected Case SQL:-

```
SELECT ac3.pk_account_id,
       ac3.vhr_account_code,
       ac3.vhr_account_name,

       cu1.dbl_credit_limit,
       cu1.sin_credit_limit_based_on,

       pt1.pk_payment_terms_id,
       pt1.sin_payment_mode,
```

```

    pt1.sin_next_day_week_month,
    pt1.sin_based_on,
    pt1.vhr_payment_term_string,

    str2.vhr_currency,

    (str2.dbl_due_debit + str2.dbl_effective_debit + str2.dbl_debit) AS dbl_debit,
    (str2.dbl_due_credit + str2.dbl_effective_credit + str2.dbl_credit) AS dbl_credit,
    (str2.dbl_due_outstanding_debit + str2.dbl_effective_outstanding_debit + str2.dbl_outstanding_debit) AS db
l_outstanding_debit,
    (str2.dbl_due_outstanding_credit + str2.dbl_effective_outstanding_credit + str2.dbl_outstanding_credit AS
dbl_outstanding_credit,

    (str2.dbl_due_debit + str2.dbl_effective_debit) AS dbl_effective_debit,
    (str2.dbl_due_credit + str2.dbl_effective_credit) AS dbl_effective_credit,
    (str2.dbl_due_outstanding_debit + str2.dbl_effective_outstanding_debit) AS dbl_effective_outstanding_debit
,
    (str2.dbl_due_outstanding_credit + str2.dbl_effective_outstanding_credit) AS dbl_effective_outstanding_cre
dit,

    str2.dbl_due_debit,
    str2.dbl_due_credit,
    str2.dbl_due_outstanding_debit,
    str2.dbl_due_outstanding_credit
FROM (
    SELECT fk_ledger_id,
           vhr_base_currency AS vhr_currency,

           SUM(dbl_base_cur_debit) AS dbl_debit,
           SUM(dbl_base_cur_credit) AS dbl_credit,
           SUM(dbl_base_cur_outstanding_debit) AS dbl_outstanding_debit,
           SUM(dbl_base_cur_outstanding_credit) AS dbl_outstanding_credit,

           SUM(dbl_base_cur_effective_debit) AS dbl_effective_debit,
           SUM(dbl_base_cur_effective_credit) AS dbl_effective_credit,
           SUM(dbl_base_cur_effective_outstanding_debit) AS dbl_effective_outstanding_debit,
           SUM(dbl_base_cur_effective_outstanding_credit) AS dbl_effective_outstanding_credit,

           SUM(dbl_base_cur_due_debit) AS dbl_due_debit,
           SUM(dbl_base_cur_due_credit) AS dbl_due_credit,
           SUM(dbl_base_cur_due_outstanding_debit) AS dbl_due_outstanding_debit,
           SUM(dbl_base_cur_due_outstanding_credit) AS dbl_due_outstanding_credit
    FROM (
        SELECT tr1.fk_ledger_id,
               tr1.vhr_base_currency,

               0.00 AS dbl_base_cur_debit,
               0.00 AS dbl_base_cur_credit,
               0.00 AS dbl_base_cur_outstanding_debit,
               0.00 AS dbl_base_cur_outstanding_credit,

               0.00 AS dbl_base_cur_effective_debit,
               0.00 AS dbl_base_cur_effective_credit,
               0.00 AS dbl_base_cur_effective_outstanding_debit,
               0.00 AS dbl_base_cur_effective_outstanding_credit,

               tr1.dbl_base_cur_debit AS dbl_base_cur_due_debit,
               tr1.dbl_base_cur_credit AS dbl_base_cur_due_credit,
               CASE WHEN tr1.dbl_base_cur_debit > 0 THEN tr1.dbl_base_cur_outstanding ELSE 0.00 END AS dbl_base_c
ur_due_outstanding_debit,
               CASE WHEN tr1.dbl_base_cur_credit > 0 THEN tr1.dbl_base_cur_outstanding ELSE 0.00 END AS dbl_base_
cur_due_outstanding_credit
        FROM transaction.tbl_transaction tr1
        INNER JOIN accounts.tbl_account ac1
            ON tr1.fk_ledger_id = ac1.pk_account_id
        WHERE tr1.dat_transaction <= <AS ON DATE>
            AND tr1.fk_main_ledger_id IN (1,2)
            AND tr1.fk_branch_id = 1
            AND tr1.fk_department_id = 1
            AND tr1.vhr_sys_module_name IN/NOT IN (???)
            AND tr1.sin_posting_status = 1
            AND (ac1.sin_ac_category = 18 OR ac1.sin_if_customer = 1)

        UNION ALL

```

```

SELECT tr2.fk_ledger_id,
       tr2.vhr_base_currency,

       0.00 AS dbl_base_cur_debit,
       0.00 AS dbl_base_cur_credit,
       0.00 AS dbl_base_cur_outstanding_debit,
       0.00 AS dbl_base_cur_outstanding_credit,

       tr2.dbl_base_cur_debit AS dbl_base_cur_effective_debit,
       tr2.dbl_base_cur_credit AS dbl_base_cur_effective_credit,
       CASE WHEN tr1.dbl_base_cur_debit > 0 THEN tr1.dbl_base_cur_outstanding ELSE 0.00 END AS dbl_base_c
ur_effective_outstanding_debit,
       CASE WHEN tr1.dbl_base_cur_credit > 0 THEN tr1.dbl_base_cur_outstanding ELSE 0.00 END AS dbl_base_
cur_effective_outstanding_credit,

       0.00 AS dbl_base_cur_due_debit,
       0.00 AS dbl_base_cur_due_credit,
       0.00 AS dbl_base_cur_due_outstanding_debit,
       0.00 AS dbl_base_cur_due_outstanding_credit
FROM transaction.tbl_transaction tr2
     INNER JOIN accounts.tbl_account ac2
       ON tr2.fk_ledger_id = ac2.pk_account_id
WHERE (tr2.dat_transaction > <AS ON DATE> AND tr2.dat_transaction <= <Effective Date>)
     AND tr2.fk_main_ledger_id IN (1,2)
     AND tr2.fk_branch_id = 1
     AND tr2.fk_department_id = 1
     AND tr2.vhr_sys_module_name IN/NOT IN (???)
     AND tr2.sin_posting_status = 1
     AND (ac2.sin_ac_category = 18 OR ac2.sin_if_customer = 1)

UNION ALL

SELECT tr3.fk_ledger_id,
       tr3.vhr_base_currency,

       tr3.dbl_base_cur_debit,
       tr3.dbl_base_cur_credit,
       CASE WHEN tr3.dbl_base_cur_debit > 0 THEN tr3.dbl_base_cur_outstanding ELSE 0.00 END AS dbl_base_c
ur_outstanding_debit,
       CASE WHEN tr3.dbl_base_cur_credit > 0 THEN tr3.dbl_base_cur_outstanding ELSE 0.00 END AS dbl_base_
cur_outstanding_credit,

       0.00 AS dbl_base_cur_effective_debit,
       0.00 AS dbl_base_cur_effective_credit,
       0.00 AS dbl_base_cur_effective_outstanding_debit,
       0.00 AS dbl_base_cur_effective_outstanding_credit,

       0.00 AS dbl_base_cur_due_debit,
       0.00 AS dbl_base_cur_due_credit,
       0.00 AS dbl_base_cur_due_outstanding_debit,
       0.00 AS dbl_base_cur_due_outstanding_credit
FROM transaction.tbl_transaction tr3
     INNER JOIN accounts.tbl_account ac3
       ON tr3.fk_ledger_id = ac3.pk_account_id
WHERE tr3.dat_transaction > <Effective Date>
     AND tr3.fk_main_ledger_id IN (1,2)
     AND tr3.fk_branch_id = 1
     AND tr3.fk_department_id = 1
     AND tr3.vhr_sys_module_name IN/NOT IN (???)
     AND tr3.sin_posting_status = 1
     AND (ac3.sin_ac_category = 18 OR ac3.sin_if_customer = 1)) AS str1
GROUP BY fk_ledger_id, vhr_base_currency) AS str2

INNER JOIN accounts.tbl_account ac3
  ON str2.fk_ledger_id = ac3.pk_account_id
INNER JOIN accounts.tbl_customer cul
  ON ac3.pk_account_id = cul.fk_account_id
LEFT JOIN accounts.tbl_payment_terms AS pt1
  ON cul.fk_payment_terms_id = pt1.pk_payment_terms_id
WHERE cul.dbl_credit_limit > 0
     AND (cul.sin_credit_limit_based_on = 2 OR str2.vhr_currency = ac3.vhr_default_currency_code)
     AND (str2.dbl_due_debit > 0.00 OR str2.dbl_due_credit > 0.00)
ORDER BY ac3.vhr_account_name;

```

**Currency = Not Base1(...) Selected Case SQL:-**

```

SELECT ac3.pk_account_id,
       ac3.vhr_account_code,
       ac3.vhr_account_name,

       cul.dbl_credit_limit,
       cul.sin_credit_limit_based_on,

       pt1.pk_payment_terms_id,
       pt1.sin_payment_mode,
       pt1.sin_next_day_week_month,
       pt1.sin_based_on,
       pt1.vhr_payment_term_string,

       str2.vhr_currency,

       ...str2.dbl_debit,
       ...str2.dbl_credit,
       ...str2.dbl_outstanding_debit,
       ...str2.dbl_outstanding_credit,

       ...

       str2.dbl_due_debit,
       str2.dbl_due_credit,
       str2.dbl_due_outstanding_debit,
       str2.dbl_due_outstanding_credit
FROM (
  SELECT fk_ledger_id,
         vhr_ledg_currency AS vhr_currency,

         SUM(dbl_ledg_cur_debit) AS dbl_debit,
         SUM(dbl_ledg_cur_credit) AS dbl_credit,
         SUM(dbl_ledg_cur_outstanding_debit) AS dbl_outstanding_debit,
         SUM(dbl_ledg_cur_outstanding_credit) AS dbl_outstanding_credit,

         ...

         SUM(dbl_ledg_cur_due_debit) AS dbl_due_debit,
         SUM(dbl_ledg_cur_due_credit) AS dbl_due_credit,
         SUM(dbl_ledg_cur_due_outstanding_debit) AS dbl_due_outstanding_debit,
         SUM(dbl_ledg_cur_due_outstanding_credit) AS dbl_due_outstanding_credit
FROM (
  SELECT tr1.fk_ledger_id,
         tr1.vhr_ledg_currency,

         0.00 AS dbl_ledg_cur_debit,
         0.00 AS dbl_ledg_cur_credit,
         0.00 AS dbl_ledg_cur_outstanding_debit,
         0.00 AS dbl_ledg_cur_outstanding_credit,

         tr1.dbl_ledg_cur_debit AS dbl_ledg_cur_due_debit,
         tr1.dbl_ledg_cur_credit AS dbl_ledg_cur_due_credit,
         CASE WHEN tr1.dbl_ledg_cur_debit > 0 THEN tr1.dbl_ledg_cur_outstanding ELSE 0.00 END AS dbl_ledg_c
ur_due_outstanding_debit,
         CASE WHEN tr1.dbl_ledg_cur_credit > 0 THEN tr1.dbl_ledg_cur_outstanding ELSE 0.00 END AS dbl_ledg_
cur_due_outstanding_credit
FROM transaction.tbl_transaction tr1
  INNER JOIN accounts.tbl_account ac1
    ON tr1.fk_ledger_id = ac1.pk_account_id
WHERE tr1.dat_transaction <= <AS ON DATE>
  AND tr1.fk_main_ledger_id IN (1,2)
  AND tr1.fk_branch_id = 1
  AND tr1.fk_department_id = 1
  AND tr1.sin_posting_status = 1
    AND tr1.vhr_sys_module_name IN/NOT IN (???)
  AND tr1.vhr_ledg_currency = '???'
  AND (ac1.sin_ac_category = 18 OR ac1.sin_if_customer = 1)

UNION ALL

SELECT tr2.fk_ledger_id,
       tr2.vhr_base_currency, ---**1->change to vhr_ledg_currency

```

```

0.00 AS dbl_ledg_cur_debit,
0.00 AS dbl_ledg_cur_credit,
0.00 AS dbl_ledg_cur_outstanding_debit,
0.00 AS dbl_ledg_cur_outstanding_credit,

tr2.dbl_ledg_cur_debit AS dbl_ledg_cur_effective_debit,
tr2.dbl_ledg_cur_credit AS dbl_ledg_cur_effective_credit,
CASE WHEN tr1.dbl_ledg_cur_debit > 0 THEN tr1.dbl_ledg_cur_outstanding ELSE 0.00 END AS dbl_ledg_c
ur_effective_outstanding_debit,
CASE WHEN tr1.dbl_ledg_cur_credit > 0 THEN tr1.dbl_ledg_cur_outstanding ELSE 0.00 END AS dbllledg_c
ur_effective_outstanding_credit,

0.00 AS dbl_ledg_cur_due_debit,
0.00 AS dbl_ledg_cur_due_credit,
0.00 AS dbl_ledg_cur_due_outstanding_debit,
0.00 AS dbl_ledg_cur_due_outstanding_credit
FROM transaction.tbl_transaction tr2
INNER JOIN accounts.tbl_account ac2
ON tr2.fk_ledger_id = ac2.pk_account_id
WHERE (tr2.dat_transaction > <AS ON DATE> AND tr2.dat_transaction <= <Effective Date>)
AND tr2.fk_main_ledger_id IN (1,2)
AND tr2.fk_branch_id = 1
AND tr2.fk_department_id = 1
AND tr2.vhr_sys_module_name IN/NOT IN (???)
AND tr2.sin_posting_status = 1
AND (ac2.sin_ac_category = 18 OR ac2.sin_if_customer = 1)

UNION ALL

SELECT tr3.fk_ledger_id,
tr3.vhr_ledg_currency,

tr3.dbl_ledg_cur_debit,
tr3.dbl_ledg_cur_credit,
CASE WHEN tr3.dbl_ledg_cur_debit > 0 THEN tr3.dbl_ledg_cur_outstanding ELSE 0.00 END AS dbl_ledg_c
ur_outstanding_debit,
CASE WHEN tr3.dbl_ledg_cur_credit > 0 THEN tr3.dbl_ledg_cur_outstanding ELSE 0.00 END AS dbl_ledg_c
ur_outstanding_credit,

0.00 AS dbl_ledg_cur_due_debit,
0.00 AS dbl_ledg_cur_due_credit,
0.00 AS dbl_ledg_cur_due_outstanding_debit,
0.00 AS dbl_ledg_cur_due_outstanding_credit
FROM transaction.tbl_transaction tr3
INNER JOIN accounts.tbl_account ac3
ON tr3.fk_ledger_id = ac3.pk_account_id
WHERE tr3.dat_transaction > <Effective Date>
AND tr3.fk_main_ledger_id IN (1,2)
AND tr3.fk_branch_id = 1
AND tr3.fk_department_id = 1
AND tr3.vhr_sys_module_name IN/NOT IN (???)
AND tr3.sin_posting_status = 1
AND tr3.vhr_ledg_currency = '???'
AND (ac3.sin_ac_category = 18 OR ac3.sin_if_customer = 1)) AS str1
GROUP BY fk_ledger_id, vhr_ledg_currency) AS str2

INNER JOIN accounts.tbl_account ac3
ON str2.fk_ledger_id = ac3.pk_account_id
INNER JOIN accounts.tbl_customer cul
ON ac3.pk_account_id = cul.fk_account_id
LEFT JOIN accounts.tbl_payment_terms AS pt1
ON cul.fk_payment_terms_id = pt1.pk_payment_terms_id
WHERE cul.dbl_credit_limit > 0
AND (cul.sin_credit_limit_based_on = 1 AND str2.vhr_currency = ac3.vhr_default_currency_code)
AND (str2.dbl_due_debit > 0.00 OR str2.dbl_due_credit > 0.00)
ORDER BY ac3.vhr_account_name;

```

#### #4 - 05/25/2023 05:30 PM - Anil KV

##### CS Coding Logic

1) Query Payment teams is null then take GUI Payment teams BASED ON Checkbox ON, Otherwise Apply Input payment team for all customers

1.1) Pass to API GUI Payment teams **Payment Mode** and **Days**

1.2) Due days calculate -> DueDate = GlobalFunctions.FnGetPaymentDueDate -> DueDate - Effective date(Input)

2) Find Payment teams and Effective date base Due days and Due amount Calculate

3) Effective Date > As on date - other wise block

4) **Outstanding** Column Value is **dbl\_due\_outstanding\_debit-dbl\_due\_outstanding\_credit**(If Credit Checkbox ON)-After As on Date matched amount(If checkbox OFF)

**#5 - 05/31/2023 04:15 PM - Anil KV**

- Status changed from New to Open
- % Done changed from 0 to 10

**#6 - 06/06/2023 11:32 AM - Junaid M**

- Status changed from Open to Ready for Coding
- Assignee changed from Junaid M to Amal Siby
- % Done changed from 10 to 20

**#7 - 06/07/2023 04:32 PM - Amal Siby**

- Status changed from Ready for Coding to Coding Started
- Start date changed from 02/02/2023 to 02/07/2023
- % Done changed from 20 to 40

**#8 - 07/07/2023 06:48 PM - Amal Siby**

- Status changed from Coding Started to Particaly Coding Done

Pending To Do  
-----

Search to Do based on( Credit Limit, Credit Limit based On) According to New Changed In Customer

**#9 - 07/10/2023 05:02 PM - unnikannan S**

- Status changed from Particaly Coding Done to Reopen Bugs
- % Done changed from 40 to 50
- Tested By set to unnikannan S

ISSUE FOUNDED

- 1.UI CRASHING WHILE SELECTING MODULE
- 2.DROP DOWN ICON MISSING WHILE SELECTING ALL IN MODULES
- 3.WHILE CHOOSING CURRENCY SHOWING OPTION US 2 MORE

**#10 - 08/02/2023 06:48 PM - Amal Siby**

Bug Fixed.

Pending To Do  
-----

Search to Do based on( Credit Limit, Credit Limit based On) According to New Changed In Customer  
Module Sorting.

**#11 - 08/20/2023 12:18 PM - Anil KV**

--\*\*\*1->change to vhr\_ledg\_currency

**#12 - 08/21/2023 06:59 PM - Amal Siby**

- Status changed from Reopen Bugs to Reopen Bugs Coding Done
- % Done changed from 50 to 60

Module Completed.

**#13 - 11/06/2023 03:10 PM - Seema Mathew**

- Status changed from Reopen Bugs Coding Done to Reopen Bugs
- % Done changed from 60 to 50
- Tested By changed from unnikannan S to Seema Mathew

ISSUE FOUNDED\*\*\*\*

- 1.PAYMENT TERMS DATA MUST BE IN PRINT.
- 2.MISSING ERASER FILTER TAB.

**#14 - 11/07/2023 08:24 PM - Amal Siby**

- Status changed from Reopen Bugs to Reopen Bugs Coding Done
- % Done changed from 50 to 60

**#15 - 11/11/2023 11:26 AM - unnikannan S**

ISSUE FOUNDED

- 1.SHOW ACT US CLIENT A/C'S DATA (BASED ON CONSIDER FILTER)
- 2.SHOW DATA BASED ON SELECTING PAYMENT TERMS ITEMS WITHOUT CHECK BOX

**#16 - 01/16/2024 12:51 PM - unnikannan S**

- Status changed from Reopen Bugs Coding Done to Reopen Bugs
- % Done changed from 60 to 50
- Tested By changed from Seema Mathew to unnikannan S

ISSUE FOUNDED

- 1.SHOW DATA AS PER CHOOSEN PAYMENT TERMS BY CUSTOMER ONLY
- 2.EFFECTIVE DATE IS MANDATORY
- 3.OUTSTANDING=UP TO AS ON DATE:NET OUTSTANDING
- 4.ADD A BLANK OPTION FOR PAYMENT TERMS FIELD
- 5.REDUCE CURRENCY COLUMN WIDTH IN LIST

**#17 - 03/07/2024 04:22 PM - Anil KV**

- Assignee changed from Amal Siby to Anil KV

**#18 - 03/07/2024 04:24 PM - Anil KV**

- Assignee changed from Anil KV to Athul P

**#19 - 03/07/2024 04:24 PM - Anil KV**

- Assignee changed from Athul P to Amal Siby

**#20 - 03/07/2024 04:25 PM - Anil KV**

- Assignee changed from Amal Siby to Anil KV

**#21 - 03/09/2024 05:57 PM - Anil KV**

**New Change**

```
=> Bill Submission - checkbox default untick
=> Payment Terms - checkbox default untick
=> Currency Default - Ledger Currency
=> Payment Terms Default Blank
=> Bill Submission - selectbox(2nd) - default blank
=> popup LedgerSummaryPupup - list Icons click
=> Bill Submission
=> Select 'As On Date' then disable and Empty - Bill Submission(Multiselectbox, Selectbox and Checkbox) - Red
*
=> Select 'Base Date' then enable Bill Submission(Multiselectbox-Tick all, Selectbox(Empty) and Checkbox((Unti
ck))) - Remove *
=> 'Base Date*' set last month end date - default, Effactive date - Current date
=> Add Customer input(Code and Name)

=> SQL and Logic
--'Base Date' or select a Bill Submission Term No and all customer then create each 'Bill Submission Term' Bil
l Submission Date(<='Base Date') list [(Bill Submission Term No, Bill Submission Date), ...]
--'As On Date' then create a 'Bill Submission Term' Bill Submission Date(='As On Date') list [(NOT NULL, 'As O
n Date')]
--LOOP ALL [(Bill Submission Term No, Bill Submission Date), ...]
-- UNION - IF Not empty
    SELECT ac.pk_account_id,
    MAX(ac.vhr_account_code) AS vhr_account_code,
```

```

MAX(ac.vhr_account_name) AS vhr_account_name,

--Bill Submission Date::Date(input string to date object) AS dat_bill_submission
MAX(cus.fk_payment_terms_id) AS fk_payment_terms_id,
MAX(cus.sin_bill_submission_term) AS sin_bill_submission_term,
--MAX(cus.sin_no_of_submission_days) AS sin_no_of_submission_days,
MAX(cus.sin_grace_period) AS sin_grace_period,
MAX(exd.sin_extra_days) AS sin_extra_days,

MAX(tr.vhr_ledg_currency) AS vhr_ledg_currency,
SUM(tr.dbl_ledg_cur_debit) AS dbl_ledg_cur_debit,
SUM(tr.dbl_ledg_cur_credit) AS dbl_ledg_cur_credit,
SUM(tr.dbl_ledg_cur_outstanding) AS dbl_ledg_cur_outstanding,-- dbl_ledg_cur_dr_outstanding, dbl_ledg_
cur_cr_outstanding

SUM(tr.dbl_base_cur_debit) AS dbl_base_cur_debit,
SUM(tr.dbl_base_cur_credit) AS dbl_base_cur_credit,
SUM(tr.dbl_base_cur_outstanding ) AS dbl_base_cur_outstanding -- dbl_base_cur_dr_outstanding, dbl_base
_cur_cr_outstanding

FROM transaction.tbl_transaction tr
INNER JOIN accounts.tbl_account ac
ON tr.fk_ledger_id = ac.pk_account_id
INNER JOIN accounts.tbl_customer cus
ON ac.pk_account_id = cus.fk_account_id
LEFT JOIN (SELECT fk_account_id,
SUM(sin_extra_days) AS sin_extra_days
FROM creditlimit.tbl_extra_limit
WHERE dat_effective_from >= '2024-03-01' AND dat_effective_to <= '2024-03-30' --'2024-03-01'->
Bill Date, '2024-03-30'->Effective Date
GROUP BY fk_account_id) exd
ON ac.pk_account_id = exd.fk_account_id
WHERE cus.fk_payment_terms_id IS NOT NULL --IF INPUT THEN Not need
AND cus.sin_bill_submission_term = 1 --IF INPUT and ALL CUSTOMER then Not need
--AND tr.dat_transaction <= Bill Submission Date
--Input based WHERE add
GROUP BY pk_account_id, vhr_ledg_currency -- vhr_ledg_currency not need Currency != Ledger Currency
--IF Calculation Based On == Outstanding
-- Consider considerMatchingDoneAfterEffectiveDate

```

## #22 - 03/11/2024 05:36 PM - Anil KV

- Assignee changed from Anil KV to Athul P

## #23 - 05/08/2024 10:12 AM - Junaid M

- Status changed from Reopen Bugs to New Changes

## #24 - 08/12/2024 09:50 AM - Athul P

- Status changed from New Changes to Coding Started

- % Done changed from 50 to 30

## #25 - 08/14/2024 10:57 AM - Athul P

- Status changed from Coding Started to Coding Done

- % Done changed from 30 to 60

## #26 - 08/16/2024 09:27 AM - unnikannan S

- Status changed from Coding Done to Testing Started

## #27 - 08/16/2024 10:01 AM - unnikannan S

- Status changed from Testing Started to Reopen Bugs

- % Done changed from 60 to 50

## ISSUE FOUNDED

- 1.Error While Search With Bill Date
- 2.Syntax Error While Changing Calculation Based On
- 3.Missing Main Ledger While Focus Out From Customer Code
- 4.Error While Clearing Default Value From Bill Submission Date No



**#28 - 08/20/2024 12:08 PM - Athul P**

- Status changed from Reopen Bugs to Reopen Bugs Coding Done
- % Done changed from 50 to 70

**#29 - 08/21/2024 10:27 AM - Arathy PS**

- Status changed from Reopen Bugs Coding Done to Reopen Bugs
- % Done changed from 70 to 50

Issue found:

1)Data is appearing when all the values are selected to exclude from the module label

**#30 - 09/05/2024 05:34 PM - Athul P**

- Status changed from Reopen Bugs to Reopen Bugs Coding Done
- % Done changed from 50 to 70

**#31 - 09/30/2024 07:17 PM - unnikannan S**

- Status changed from Reopen Bugs Coding Done to Testing Done
- % Done changed from 70 to 90

**#32 - 10/02/2024 10:43 AM - Anonymous**

- Assignee changed from Athul P to Junaid M

**#33 - 10/03/2024 05:41 PM - Junaid M**

- Status changed from Testing Done to New Changes
- % Done changed from 90 to 50

Workflow Changes [03/10/2024]

Customer, Supplier Master:

1. In System Tab -> Credit Limit Based On -> Add validation, If "Credit Limit Amount" is given then "Credit Limit Based On" is mandatory.

Extra Credit Limit/Payment Days

1. Split Extra Limit widget. Add Currency as label on right side

Customer Overdue/Supplier/Staff Overdue Report

1. Add title if there is a star mark on widgets.
2. Add blank option in currency select box In Search filter area
3. Workflow change:- -> If Chooses Currency = "Ledger Currency" then only show overdue of accounts in which credit limit is set based on "Account Currency"
  - > If Chooses Currency = "Base Currency" then only show overdue of accounts in which credit limit is set based on "Base Currency"
  - > If Chooses Currency = "Individual Currency" then only show overdue of accounts in which transaction currency is selected Currency.
  - > If Chooses Currency = "Blank Currency" option then only show overdue of accounts in all currency

**#34 - 11/25/2024 11:08 AM - Arathy PS**

- Assignee changed from Junaid M to Vivekp V

**#35 - 11/26/2024 11:41 AM - Arathy PS**

- Status changed from New Changes to Reopen Bugs

NEW CHANGES DONE

1. Add an additional column 'Total Due Amount' with title: total due upto the end of bill period, also add title for due amount.
2. Add an option on the constrain: 'Overdue based on total due amount'(default tick) add title.

ISSUE FIXED

1. Data showing when calculation based on: 'debit & credit outstanding net' or 'debit & credit net' is selected and only credit entry is present for the

selected as on and effective date then data is showing.

**#36 - 11/28/2024 09:24 AM - Vivekp V**

- Status changed from Reopen Bugs to Reopen Bugs Coding Done
- % Done changed from 50 to 70

**#37 - 01/02/2025 11:25 AM - Arathy PS**

- Status changed from Reopen Bugs Coding Done to Reopen Bugs
- % Done changed from 70 to 50

ISSUE FIXED

- 1. Not filtering the data based on payment terms.
- 2. Not taking data from sale tab module.

**#38 - 01/02/2025 02:59 PM - Vivekp V**

- Status changed from Reopen Bugs to Reopen Bugs Coding Done
- % Done changed from 50 to 70

**#39 - 01/03/2025 12:19 PM - Arathy PS**

- Status changed from Reopen Bugs Coding Done to New Changes
- Assignee changed from Vivekp V to Augustin Jose
- % Done changed from 70 to 50

NEW CHANGES

- 1. The data in 'calculation based on' field should appear based on the admin settings.

**#40 - 01/07/2025 10:46 AM - Augustin Jose**

- Status changed from New Changes to New Changes Coding Done
- % Done changed from 50 to 70

**#41 - 01/13/2025 11:40 AM - Arathy PS**

- Status changed from New Changes Coding Done to New Changes
- % Done changed from 70 to 50

New Changes

- 1. Three new fields are added in the payment terms--Submission Date, Corporate Approved (Submission), Corporate Approved (Invoice).
- Make corresponding changes here also
- Discuss with vivek and anil

**#42 - 02/22/2025 03:19 PM - Augustin Jose**

- Status changed from New Changes to New Changes Coding Done
- % Done changed from 50 to 70

**#43 - 03/06/2025 12:16 PM - Theja Ponon**

- File clipboard-202503061214-67j2u.png added
- File clipboard-202503061214-mw7gz.png added
- File clipboard-202503061216-hbm3d.png added
- Status changed from New Changes Coding Done to Reopen Bugs
- % Done changed from 70 to 50

Issue Found

- 1)Only showing single customer when same payment terms set for two customers
- 2)Incorrect value in Bill period

Travvise

RAVEL SOLUTION PVT LTD

Customer Overdue Report

As On Date\*15/02/2025

Effective\*01/03/2025

Customer\*

Bill Submission\*Days6 more

Payment Terms\*Within Next

Branch\*Main

DepartmentAccount

Customer Overdue Report

#	Code	Name	Bill Submission	Bill Period	Payment Terms	Payment Date	Grace Days*	Due Date*	Over Due Days	Currency	Last Bill Due*	Total Due Amount*
1	CU5001	Customer	Weekly(Sun-Sat)	02/02/2025 - 08/02/2...	Within Next Fortnight Based On Submission Date	28/02/2025	0	28/02/2025	1	QAR	0.00	1000.00

#44 - 03/10/2025 07:52 PM - Augustin Jose

- Status changed from Reopen Bugs to Reopen Bugs Coding Done
- % Done changed from 50 to 70

#45 - 03/20/2025 12:00 PM - Theja Ponon

- File clipboard-202503201200-35pfp.png added
- Status changed from Reopen Bugs Coding Done to Reopen Bugs
- % Done changed from 70 to 50

Issue Fixed

1)Data is not getting when giving as on date at the end of bill period

Travvise

RVL SOLUTION PVT LTD

Customer Overdue Report

As On Date\*

01/01/2025

Effective\*

20/03/2025

Customer \*

Bill Submission\*

Days(Start Date 1st)

Payment Terms\*

Within Next Month

Branch\*

Main

Department

Account

Customer Overdue Report

#	Code	Name	Bill Submission	Bill Period	Payment Terms	Payment Date	Grace Days*	Due Date*	Over Due Days	Currency	Last Bill Due*	Total Due Amount*
1	CUS003	Customer c	Fortnight	01/01/2025 - 15/01/2025	Within Next Month Based On Transaction Date	28/02/2025	0	28/02/2025	20	QAR	1000.00	1000.00

#46 - 03/20/2025 12:59 PM - Augustin Jose

- Status changed from Reopen Bugs to Reopen Bugs Coding Done
- % Done changed from 50 to 70

Files

clipboard-202503061214-67j2u.png	96.6 KB	03/06/2025	Theja Ponon
clipboard-202503061214-mw7gz.png	96.6 KB	03/06/2025	Theja Ponon
clipboard-202503061216-hbm3d.png	96 KB	03/06/2025	Theja Ponon
clipboard-202503201200-35pfp.png	72.1 KB	03/20/2025	Theja Ponon