

EOFFICE - GD4 - Bug #1590

UAT.25.03.2025

03/25/2025 11:31 AM - Luc Tran Van

Status:	New	Start date:	03/25/2025
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Description			

History

#1 - 03/25/2025 11:33 AM - Luc Tran Van

```
-- Bước 1: Lấy các bản ghi có CREATE_DT mới nhất cho mỗi REQ_ID
WITH LatestRecords AS (
SELECT
FROM [dbo].[CM_APPROVE_GROUP] a
WHERE CREATE_DT = (
SELECT MAX
FROM [dbo].[CM_APPROVE_GROUP] b
WHERE b.REQ_ID = a.REQ_ID
)
),
-- Bước 2: Đếm số lần lặp của mỗi STEP_LEVEL
StepCounts AS (
SELECT
REQ_ID,
STEP_LEVEL,
COUNT() AS StepRepeatCount
FROM LatestRecords
GROUP BY REQ_ID, STEP_LEVEL
),
-- Bước 3: Tạo chi tiết bước và thông tin tổng quan cho mỗi REQ_ID
ReqDetails AS (
SELECT
REQ_ID,
STEP_LEVEL,
COUNT() AS RepeatCount
FROM LatestRecords
GROUP BY REQ_ID, STEP_LEVEL
),
ReqSummary AS (
SELECT
r.REQ_ID,
COUNT() AS DistinctSteps, -- Đếm số bước khác nhau
(SELECT COUNT FROM LatestRecords a WHERE a.REQ_ID = r.REQ_ID) AS TotalRows,
STRING_AGG(CAST + '(' + CAST + ')', ', ' )
WITHIN GROUP (ORDER BY r.STEP_LEVEL) AS StepDetails,
(SELECT STRING_AGG(APPROVE_GROUP_ID, ', ' ) WITHIN GROUP (ORDER BY APPROVE_GROUP_ID)
FROM LatestRecords b
WHERE b.REQ_ID = r.REQ_ID) AS ApproveGroupList
FROM ReqDetails r
GROUP BY r.REQ_ID
),
-- Bước 4: Lọc các REQ_ID có STEP_LEVEL lặp lại bất thường
InvalidReq AS (
SELECT
rs.REQ_ID,
rs.DistinctSteps,
rs.TotalRows,
rs.StepDetails,
rs.ApproveGroupList,
MAX AS MaxRepeatCount
FROM ReqSummary rs
LEFT JOIN StepCounts sc ON rs.REQ_ID = sc.REQ_ID
GROUP BY rs.REQ_ID, rs.DistinctSteps, rs.TotalRows, rs.StepDetails, rs.ApproveGroupList
```

```

HAVING MAX > rs.DistinctSteps + 1 -- Lọc nếu bước lặp lại vượt quá số bước khác nhau + 1
)
SELECT
REQ_ID,
DistinctSteps AS 'Số bước khác nhau',
TotalRows AS 'Tổng số bản ghi',
StepDetails AS 'Chi tiết các bước',
ApproveGroupList AS 'Danh sách APPROVE_GROUP_ID'
FROM InvalidReq
ORDER BY REQ_ID;

```

#2 - 03/25/2025 11:43 AM - Luc Tran Van

```

-- Bước 1: Lấy các bản ghi có CREATE_DT mới nhất cho mỗi REQ_ID
WITH LatestRecords AS (
SELECT
FROM [dbo].[CM_APPROVE_GROUP] a
WHERE CREATE_DT = (
SELECT MAX
FROM [dbo].[CM_APPROVE_GROUP] b
WHERE b.REQ_ID = a.REQ_ID
)
),
-- Bước 2: Kiểm tra các REQ_ID có STEP_LEVEL không liên tục
StepCheck AS (
SELECT
REQ_ID,
MIN AS MinStep,
MAX AS MaxStep,
COUNT AS DistinctSteps,
COUNT() AS TotalRows,
STRING_AGG(CAST, ',') WITHIN GROUP (ORDER BY STEP_LEVEL) AS StepSequence,
STRING_AGG(APPROVE_GROUP_ID, ',') WITHIN GROUP (ORDER BY APPROVE_GROUP_ID) AS ApproveGroupList
FROM LatestRecords
GROUP BY REQ_ID
)
SELECT
REQ_ID,
DistinctSteps AS 'Số bước khác nhau',
TotalRows AS 'Tổng số bản ghi',
StepSequence AS 'Dãy STEP_LEVEL',
ApproveGroupList AS 'Danh sách APPROVE_GROUP_ID'
FROM StepCheck
WHERE DistinctSteps < (MaxStep - MinStep + 1) -- Lọc các REQ_ID thiếu bước
ORDER BY REQ_ID;

```

#3 - 03/25/2025 11:54 AM - Luc Tran Van

```

WITH LatestRecords AS (
SELECT
FROM [dbo].[CM_APPROVE_GROUP] a
WHERE CREATE_DT = (
SELECT MAX
FROM [dbo].[CM_APPROVE_GROUP] b
WHERE b.REQ_ID = a.REQ_ID
)
),
StepCheck AS (
SELECT
REQ_ID,
MIN AS MinStep,
MAX AS MaxStep,
COUNT AS DistinctSteps,
COUNT() AS TotalRows,
STUFF((
SELECT ',' + CAST
FROM LatestRecords b
WHERE b.REQ_ID = a.REQ_ID
ORDER BY b.STEP_LEVEL
FOR XML PATH
), 1, 1, '') AS StepSequence,
STUFF((
SELECT ',' + APPROVE_GROUP_ID
FROM LatestRecords b
WHERE b.REQ_ID = a.REQ_ID

```

```

ORDER BY b.STEP_LEVEL
FOR XML PATH
), 1, 1, ") AS ApproveGroupList
FROM LatestRecords a
GROUP BY REQ_ID
)
SELECT
REQ_ID,
DistinctSteps AS 'Số bước khác nhau',
TotalRows AS 'Tổng số bản ghi',
StepSequence AS 'Dãy STEP_LEVEL'
FROM StepCheck
WHERE DistinctSteps < (MaxStep - MinStep + 1)
ORDER BY REQ_ID DESC;

```

#4 - 03/25/2025 01:30 PM - Luc Tran Van

```

DECLARE @REQ_ID NVARCHAR
DECLARE @DistinctSteps INT
DECLARE @TotalRows INT
DECLARE @StepSequence NVARCHAR
DECLARE @MinStep INT
DECLARE @MaxStep INT

-- Tạo bảng tạm để lưu kết quả
CREATE TABLE #StepCheckResults (
REQ_ID NVARCHAR,
DistinctSteps INT,
TotalRows INT,
StepSequence NVARCHAR
)

-- Khai báo con trỏ
DECLARE StepCheckCursor CURSOR FOR
WITH LatestRecords AS (
SELECT
FROM [dbo].[CM_APPROVE_GROUP] a
WHERE CREATE_DT = (
SELECT MAX
FROM [dbo].[CM_APPROVE_GROUP] b
WHERE b.REQ_ID = a.REQ_ID
)
),
StepCheck AS (
SELECT
REQ_ID,
MIN AS MinStep,
MAX AS MaxStep,
COUNT AS DistinctSteps,
COUNT() AS TotalRows,
STUFF((
SELECT ',' + CAST
FROM LatestRecords b
WHERE b.REQ_ID = a.REQ_ID
ORDER BY b.STEP_LEVEL
FOR XML PATH
), 1, 1, ") AS StepSequence
FROM LatestRecords a
GROUP BY REQ_ID
)
SELECT TOP 2
REQ_ID,
DistinctSteps,
TotalRows,
StepSequence,
MinStep,
MaxStep
FROM StepCheck
WHERE DistinctSteps < (MaxStep - MinStep + 1)
--AND EXISTS
ORDER BY REQ_ID DESC

-- Mở con trỏ
OPEN StepCheckCursor

```

```

-- Fetch dòng đầu tiên
FETCH NEXT FROM StepCheckCursor
INTO @REQ_ID, @DistinctSteps, @TotalRows, @StepSequence, @MinStep, @MaxStep

-- Vòng lặp xử lý con trỏ
WHILE @@FETCH_STATUS = 0
BEGIN
-- In ra thông tin chi tiết của từng REQ_ID
PRINT 'REQ_ID: ' + @REQ_ID
PRINT 'Số bước khác nhau: ' + CAST
PRINT 'Tổng số bản ghi: ' + CAST
PRINT 'Dãy STEP_LEVEL: ' + @StepSequence
PRINT 'Bước nhỏ nhất: ' + CAST
PRINT 'Bước lớn nhất: ' + CAST
PRINT '-----'
select @req_id
-- Khai báo bảng tạm
create table #listApproveTlb (
TEMPLATE_REQUEST_ID NVARCHAR,
STEP_LEVEL INT,
GROUP_APPROVE NVARCHAR
)

create table #approveTlb (
TEMPLATE_REQUEST_ID NVARCHAR,
STEP_LEVEL INT,
USERNAME NVARCHAR
)

-- Lấy bản ghi mới nhất cho mỗi TEMPLATE_REQUEST_ID
;WITH LatestApproveLog AS (
SELECT
TEMPLATE_REQUEST_ID,
GROUP_APPROVE,
CREATE_DT,
ROW_NUMBER() OVER (PARTITION BY TEMPLATE_REQUEST_ID ORDER BY CREATE_DT DESC) AS RowNum
FROM CM_TEMPLATE_APPROVE_LOG
WHERE TEMPLATE_REQUEST_ID = @REQ_ID
)
-- Chèn dữ liệu từ bản ghi mới nhất vào bảng tạm
INSERT INTO #listApproveTlb (TEMPLATE_REQUEST_ID, STEP_LEVEL, GROUP_APPROVE)
SELECT
TEMPLATE_REQUEST_ID,
ROW_NUMBER() OVER (PARTITION BY TEMPLATE_REQUEST_ID ORDER BY (SELECT 1)),
[value]
FROM LatestApproveLog
CROSS APPLY WSISPLIT
WHERE RowNum = 1
AND [value] IS NOT NULL
AND [value] <> ''

-- Chèn dữ liệu vào bảng username
INSERT INTO #approveTlb (TEMPLATE_REQUEST_ID, STEP_LEVEL, USERNAME)
SELECT
A.TEMPLATE_REQUEST_ID,
A.STEP_LEVEL,
B.[value]
FROM #listApproveTlb A
CROSS APPLY (
SELECT [value]
FROM WSISPLIT
) B
WHERE B.[value] IS NOT NULL AND B.[value] <> ''

-- Hiển thị kết quả cuối cùng
SELECT
A.TEMPLATE_REQUEST_ID,
A.STEP_LEVEL,
A.USERNAME
FROM #approveTlb A
select * from cm_approve_group where req_id = @req_id
--25032025_secretkey
UPDATE A SET A.STEP_LEVEL = B.STEP_LEVEL
FROM dbo.CM_APPROVE_GROUP A
INNER JOIN #approveTlb B ON A.REQ_ID = B.TEMPLATE_REQUEST_ID AND A.APPROVE_USERNAME = B.USERNAME

```

WHERE A.REQ_ID = @REQ_ID

drop table #approveTlb

drop table #listApproveTlb

-- Fetch dòng tiếp theo

```
    FETCH NEXT FROM StepCheckCursor
    INTO @REQ_ID, @DistinctSteps, @TotalRows, @StepSequence, @MinStep, @MaxStep
END
```

-- Đóng và giải phóng con trỏ

CLOSE StepCheckCursor

DEALLOCATE StepCheckCursor

--25032025_secretkey

#5 - 03/25/2025 03:06 PM - Luc Tran Van

USE [gAMSPPro_BVB_v3_FINAL1]

GO

/****** Object: StoredProcedure [dbo].[CM_REQUEST_TEMPLATE_Search] Script Date: 3/25/2025 3:05:50 PM **/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[CM_REQUEST_TEMPLATE_Search]

@p_REQUEST_TEMPLATE_CODE VARCHAR=NULL,

@p_REQUEST_TEMPLATE_NAME NVARCHAR=NULL,

@p_NOTES NVARCHAR=NULL,

@p_RECORD_STATUS VARCHAR=NULL,

@p_MAKER_ID VARCHAR=NULL,

@p_CREATE_DT VARCHAR=NULL,

@p_AUTH_STATUS VARCHAR=NULL,

@p_CHECKER_ID VARCHAR=NULL,

@p_APPROVE_DT VARCHAR=NULL,

@p_MaxResultCount INT = NULL,

@p_SkipCount INT = NULL,

@p_Sorting VARCHAR = NULL,

@p_TotalCount INT = NULL,

@p_REPORT_NO NVARCHAR=NULL,

@p_REPORT_DT NVARCHAR=NULL,

@p_TYPE_TEMPLATE NVARCHAR=NULL,

@p_FROM_DT NVARCHAR=NULL,

@p_TO_DT NVARCHAR=NULL,

@p_TYPE_TEMPLATE_ID VARCHAR=NULL,

@p_MAKER_FULLNAME NVARCHAR=NULL,

@p_USER_LOGIN VARCHAR=NULL,

@p_AUTH_STATUS_APP VARCHAR=NULL,

@p_TITLE NVARCHAR=NULL,

@p_USER_APPROVES NVARCHAR=NULL,

@p_FROM_DATE VARCHAR=NULL,

@p_TO_DATE VARCHAR=NULL,

@p_TOP INT = NULL,

@p_BRANCH_ID VARCHAR = NULL,

@p_DEP_ID VARCHAR = NULL

AS

BEGIN -- PAGING

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

SET NOCOUNT ON;

--NGUYENTD 19082024_SECRETKEY

SELECT * INTO #LIST_ROLE FROM TL_USER_GET_ROLES(@p_USER_LOGIN)

CREATE TABLE #CM_REQUEST_TEMPLATE (

REQUEST_TEMPLATE_ID VARCHAR NOT NULL

,REQUEST_TEMPLATE_NAME NVARCHAR NULL

,RECORD_STATUS VARCHAR NULL

,MAKER_ID VARCHAR NULL

,CREATE_DT DATETIME NULL

,AUTH_STATUS VARCHAR NULL

,CHECKER_ID VARCHAR NULL

,APPROVE_DT DATETIME NULL

,CREATE_LOCATION NVARCHAR NULL

,HEADER NVARCHAR NULL

,BRANCH_ID NVARCHAR NULL

,REPORT_NO NVARCHAR NULL

,TITLE NVARCHAR NULL

,REPORT_DT DATETIME NULL

,SCHEME VARCHAR NULL

```

,TYPE_TEMPLATE_ID NVARCHAR NULL
,CURRENT_STEP INT NULL
,IS_SENT_APPROVE BIT NULL
,IS_BACK_DATE BIT NULL
)
--CREATE NONCLUSTERED INDEX idx_BR ON #CM_REQUEST_TEMPLATE (BRANCH_ID)WITH (FILLFACTOR = 80)
--CREATE NONCLUSTERED INDEX idx_REQ_ID ON #CM_REQUEST_TEMPLATE (REQUEST_TEMPLATE_ID)WITH (FILLFACTOR = 80)
--CREATE NONCLUSTERED INDEX idx_RECORD_STATUS ON #CM_REQUEST_TEMPLATE (RECORD_STATUS)WITH (FILLFACTOR = 80)
----CREATE NONCLUSTERED INDEX idx_MAKER_ID ON #CM_REQUEST_TEMPLATE (MAKER_ID)WITH (FILLFACTOR = 80)
--CREATE NONCLUSTERED INDEX idx_TYPE_TEMPLATE_ID ON #CM_REQUEST_TEMPLATE (TYPE_TEMPLATE_ID)WITH (FILLFACTOR = 80)
CREATE TABLE #AUTH_STATUS (REQ_ID VARCHAR, AUT_STATUS VARCHAR,TYPE VARCHAR)

CREATE NONCLUSTERED INDEX temp_ReqId ON #AUTH_STATUS(AUT_STATUS) WITH (FILLFACTOR = 80)
CREATE NONCLUSTERED INDEX temp_AutStatus ON #AUTH_STATUS(REQ_ID) WITH (FILLFACTOR = 80)
--DBCC SHOW_STATISTICS ('CM_DEPARTMENT', 'idx_BR_ID')

INSERT INTO #CM_REQUEST_TEMPLATE
SELECT A.REQUEST_TEMPLATE_ID, A.REQUEST_TEMPLATE_NAME, A.RECORD_STATUS, A.MAKER_ID, A.CREATE_DT, A.AUTH_STAT
US, A.CHECKER_ID, A.APPROVE_DT, A.CREATE_LOCATION,
A.HEADER, A.BRANCH_ID, A.REPORT_NO, A.TITLE, A.REPORT_DT, A.SCHEME, A.TYPE_TEMPLATE_ID, A.CURRENT_STEP, A.IS
_SENT_APPROVE, A.IS_BACK_DATE
FROM CM_REQUEST_TEMPLATE A
WHERE 1 =1
AND (@p_REQUEST_TEMPLATE_NAME IS NULL OR @p_REQUEST_TEMPLATE_NAME='' OR REQUEST_TEMPLATE_NAME LIKE '%' +
@p_REQUEST_TEMPLATE_NAME + '%')
AND (@p_TYPE_TEMPLATE_ID IS NULL OR @p_TYPE_TEMPLATE_ID='' OR A.TYPE_TEMPLATE_ID LIKE '%' + @p_TYPE_TEMP
LATE_ID + '%')
--
AND (@p_NOTES IS NULL OR @p_NOTES='' OR A.NOTES = @p_NOTES)
AND (@p_RECORD_STATUS IS NULL OR @p_RECORD_STATUS='' OR A.RECORD_STATUS = @p_RECORD_STATUS)
AND (@p_MAKER_ID IS NULL OR @p_MAKER_ID='' OR A.MAKER_ID LIKE '%' + @p_MAKER_ID + '%')
AND (@p_CREATE_DT IS NULL OR DATEDIFF(DAYOFYEAR,A.CREATE_DT,CONVERT(datetime, @p_CREATE_DT,103)) = 0)
--(@p_MAKER_ID IS NULL OR @p_MAKER_ID='' OR MAKER_ID = @p_MAKER_ID) AND
AND (@p_AUTH_STATUS IS NULL OR @p_AUTH_STATUS='' OR A.AUTH_STATUS = @p_AUTH_STATUS)
AND (@p_CHECKER_ID IS NULL OR @p_CHECKER_ID='' OR A.CHECKER_ID = @p_CHECKER_ID)
AND (@p_APPROVE_DT IS NULL OR DATEDIFF(DAYOFYEAR,A.APPROVE_DT,CONVERT(datetime, @p_APPROVE_DT,103)) = 0)
AND (A.REPORT_DT >= CONVERT(DATE,@p_FROM_DATE,103) OR @p_FROM_DATE IS NULL OR @p_FROM_DATE = '')
AND (A.REPORT_DT <= CONVERT(DATE,@p_TO_DATE,103) OR @p_TO_DATE IS NULL OR @p_TO_DATE = '')
AND (REPORT_DT=CONVERT(DATETIME,@p_REPORT_DT,103) OR @p_REPORT_DT IS NULL)
AND (@p_TITLE IS NULL OR @p_TITLE='' OR A.TITLE LIKE '%' + @p_TITLE + '%')
AND (@p_REPORT_NO IS NULL OR @p_REPORT_NO='' OR A.REPORT_NO LIKE '%' + @p_REPORT_NO + '%')
AND ((A.RECORD_STATUS = '0' AND (A.MAKER_ID = @p_USER_LOGIN)) OR A.RECORD_STATUS = '1')
ORDER BY A.CREATE_DT DESC
INSERT #AUTH_STATUS
SELECT G.REQ_ID,
CASE
WHEN DONE=0 THEN 'U'
WHEN DONE=1 THEN 'A'
END AS AUT_STATUS
,G.TYPE
FROM CM_APPROVE_GROUP G
INNER JOIN #CM_REQUEST_TEMPLATE ON REQUEST_TEMPLATE_ID = G.REQ_ID
WHERE (DONE=1 OR PROCESS_STATUS=1 OR [TYPE]='AUT') AND (@p_USER_LOGIN=APPROVE_USERNAME OR @p_USER_
LOGIN=AUTHORITY_NAME)
CREATE TABLE #APPROVE_GROUP (REQ_ID VARCHAR(15), USER_APPROVES NVARCHAR(MAX))
INSERT #APPROVE_GROUP(REQ_ID, USER_APPROVES)
select G.REQ_ID,STRING_AGG(G.APPROVE_USERNAME + ' - ' + u.TLFullName, ';') USER_APPROVES
FROM CM_APPROVE_GROUP G
JOIN TL_USER U ON G.APPROVE_USERNAME=U.TLNANME
INNER JOIN #CM_REQUEST_TEMPLATE ON G.REQ_ID = REQUEST_TEMPLATE_ID
WHERE PROCESS_STATUS=1 AND DONE=0 AND [TYPE]='APP'
GROUP BY G.REQ_ID
CREATE TABLE #SHARE (REQ_ID VARCHAR(15), SHARE_USERNAME VARCHAR(MAX))
INSERT #SHARE
(
REQ_ID,
SHARE_USERNAME
)
select REQUEST_TEMPLATE_ID,STRING_AGG(SHARE_USERNAME, ';') SHARE_USERNAME
FROM CM_TEMPLATE_SHARE
INNER JOIN #CM_REQUEST_TEMPLATE ON REQ_ID = REQUEST_TEMPLATE_ID
GROUP BY REQUEST_TEMPLATE_ID
CREATE TABLE #BRANCH (BRANCH_ID VARCHAR(15))
INSERT #BRANCH
SELECT DE.BRANCH_ID FROM CM_DEPARTMENT DE INNER JOIN #CM_REQUEST_TEMPLATE A ON DE.DEP_ID = A.BRANCH_ID GRO

```

```

UP BY DE.BRANCH_ID
    IF @P_TOP IS NULL OR @P_TOP = ''
    BEGIN

-- PAGING BEGIN
SELECT A.*,B.CONTENT AS AUTH_STATUS_NAME,C.CONTENT AS RECORD_STATUS_NAME ,TT.TYPE_TEMPLATE_NAME AS
TYPE_TEMPLATE_NAME
,CASE
WHEN BR.[BRANCH_ID] IS NOT NULL THEN BR.[BRANCH_NAME]
ELSE (SELECT TOP 1 D.[BRANCH_NAME] + '-' + C.[DEP_NAME]
FROM CM_DEPARTMENT C
LEFT JOIN CM_BRANCH D ON D.BRANCH_ID = C.BRANCH_ID
WHERE C.DEP_ID = A.BRANCH_ID)
END AS BRANCH_NAME--NGUYENTD 17032024_SECRETKEY: FIX TÌM KIẾM THEO ĐƠN VỊ
,U.TLFullName AS MAKER_FULLNAME,AUTH.CONTENT AS AUTH_STATUS_APP_NAME,GR.USER_APPROVES,SHARE.SHARE_USERNAME
-- SELECT END
FROM #CM_REQUEST_TEMPLATE A
LEFT JOIN CM_ALLCODE C ON C.CDNAME = 'RECORD_STATUS' AND C.CDVAL = A.RECORD_STATUS
LEFT JOIN CM_TYPE_TEMPLATE TT ON TT.TYPE_TEMPLATE_ID=A.TYPE_TEMPLATE_ID
LEFT JOIN #BRANCH CA ON CA.BRANCH_ID = A.BRANCH_ID --NGUYENTD 17032024_SECRETKEY: FIX TÌM KIẾM THEO ĐƠN VỊ TẠO
LEFT JOIN CM_BRANCH BR ON BR.BRANCH_ID = ca.BRANCH_ID
LEFT JOIN CM_DEPARTMENT CD ON CD.DEP_ID = A.BRANCH_ID
LEFT JOIN TL_USER U ON U.TLNANME=A.MAKER_ID
--WORKFLOW
LEFT JOIN #AUTH_STATUS AG ON AG.REQ_ID=A.REQUEST_TEMPLATE_ID AND TYPE <> 'AUT'
LEFT JOIN dbo.CM_ALLCODE AUTH ON AUTH.CDVAL = AG.AUTH_STATUS AND AUTH.CDNAME = 'AUTH_STATUS_APP'
LEFT JOIN dbo.CM_ALLCODE B ON B.CDVAL = A.AUTH_STATUS AND B.CDNAME = 'AUTH_STATUS_ONL'
LEFT JOIN #APPROVE_GROUP GR ON GR.REQ_ID = A.REQUEST_TEMPLATE_ID
--END WORKFLOW
LEFT JOIN #SHARE SHARE ON SHARE.REQ_ID = A.REQUEST_TEMPLATE_ID
WHERE 1=1
AND (@p_MAKER_FULLNAME IS NULL OR @p_MAKER_FULLNAME="" OR U.TLFullName LIKE '%'+@p_MAKER_FULLNAME+'%' OR
U.TLNANME LIKE '%'+@p_MAKER_FULLNAME+'%')
AND (@p_AUTH_STATUS_APP IS NULL OR @p_AUTH_STATUS_APP="" OR AG.AUTH_STATUS = @p_AUTH_STATUS_APP)
--WORKFLOW
AND (@p_USER_APPROVES IS NULL OR @p_USER_APPROVES="" OR GR.USER_APPROVES LIKE '%'+@p_USER_APPROVES+'%')
AND (
(A.MAKER_ID=@p_USER_LOGIN)
OR (@p_USER_LOGIN = 'baotq')--NGUYENTD 27122023_SECRETKEY: Y/C CỦA ANH BẢO
OR (EXISTS )
OR EXISTS (SELECT 1 FROM CM_TEMPLATE_SHARE WHERE SHARE_USERNAME = @p_USER_LOGIN AND REQ_ID =
A.REQUEST_TEMPLATE_ID)
OR (EXISTS 1 FROM #LIST_ROLE USERROLE WHERE USERROLE.ROLE_ID = 'ADMIN_TO_TRINH' ))
--OR (dbo.FN_CHECK_ROLE_USER(@p_USER_LOGIN,'ADMIN_TO_TRINH')=1)
OR (EXISTS (SELECT top(1) 1 FROM #LIST_ROLE USERROLE WHERE USERROLE.ROLE_ID = 'GDDV'
AND (
EXISTS 1 FROM TL_USER US WHERE US.TLNANME = A.MAKER_ID AND (USERROLE.BRANCH_ID = US.TLSUBBRID OR
USERROLE.DEPT_ID = US.DEP_ID) AND US.TLSUBBRID <> 'DV0001')
OR EXISTS 1 FROM TL_USER US WHERE US.TLNANME = A.MAKER_ID AND USERROLE.DEPT_ID = US.DEP_ID)
)
)
)
)
--NGUYENTD 17032024_SECRETKEY: FIX TÌM KIẾM THEO ĐƠN VỊ TẠO
AND (BR.BRANCH_ID LIKE '%'+ @p_BRANCH_ID + '%' OR @p_BRANCH_ID IS NULL OR @p_BRANCH_ID = "")
AND (A.BRANCH_ID LIKE '%'+ @p_DEP_ID + '%' OR @p_DEP_ID IS NULL OR @p_DEP_ID = "")
ORDER BY A.CREATE_DT DESC
--END NGUYENTD
--END WORKFLOW
-- PAGING END
END
ELSE
BEGIN
-- PAGING BEGIN
SELECT TOP (@P_TOP) A.*,B.CONTENT AS AUTH_STATUS_NAME,C.CONTENT AS RECORD_STATUS_NAME
,TT.TYPE_TEMPLATE_NAME AS TYPE_TEMPLATE_NAME,CASE
WHEN BR.[BRANCH_ID] IS NOT NULL THEN BR.[BRANCH_NAME]
ELSE (SELECT TOP 1 D.[BRANCH_NAME] + '-' + C.[DEP_NAME]
FROM CM_DEPARTMENT C
LEFT JOIN CM_BRANCH D ON D.BRANCH_ID = C.BRANCH_ID
WHERE C.DEP_ID = A.BRANCH_ID)
END AS BRANCH_NAME
,U.TLFullName AS MAKER_FULLNAME ,AUTH.CONTENT AS
AUTH_STATUS_APP_NAME,GR.USER_APPROVES,SHARE.SHARE_USERNAME
-- SELECT END

```

```

FROM #CM_REQUEST_TEMPLATE A
LEFT JOIN CM_ALLCODE C ON C.CDNAME = 'RECORD_STATUS' AND C.CDVAL = A.RECORD_STATUS
LEFT JOIN CM_TYPE_TEMPLATE TT ON TT.TYPE_TEMPLATE_ID=A.TYPE_TEMPLATE_ID
LEFT JOIN #BRANCH CA ON CA.BRANCH_ID = A.BRANCH_ID --NGUYENTD 17032024_SECRETKEY: FIX TÌM KIẾM THEO ĐƠN VỊ
LEFT JOIN CM_BRANCH BR ON BR.BRANCH_ID = ca.BRANCH_ID
LEFT JOIN CM_DEPARTMENT CD ON CD.DEP_ID = A.BRANCH_ID
LEFT JOIN TL_USER U ON U.TLNANME=A.MAKER_ID
--WORKFLOW
LEFT JOIN #AUTH_STATUS AG ON AG.REQ_ID=A.REQUEST_TEMPLATE_ID AND TYPE <> 'AUT'
LEFT JOIN dbo.CM_ALLCODE AUTH ON AUTH.CDVAL = AG.AUTH_STATUS AND AUTH.CDNAME = 'AUTH_STATUS_APP'
LEFT JOIN dbo.CM_ALLCODE B ON B.CDVAL = A.AUTH_STATUS AND B.CDNAME = 'AUTH_STATUS_ONL'
LEFT JOIN #APPROVE_GROUP GR ON GR.REQ_ID = A.REQUEST_TEMPLATE_ID
--END WORKFLOW
LEFT JOIN #SHARE SHARE ON SHARE.REQ_ID = A.REQUEST_TEMPLATE_ID
WHERE 1=1
AND (@p_MAKER_FULLNAME IS NULL OR @p_MAKER_FULLNAME="" OR U.TLFullName LIKE '%'+@p_MAKER_FULLNAME+'%' OR
U.TLNANME LIKE '%'+@p_MAKER_FULLNAME+'%')
AND (@p_AUTH_STATUS_APP IS NULL OR @p_AUTH_STATUS_APP="" OR AG.AUTH_STATUS = @p_AUTH_STATUS_APP)
--WORKFLOW
AND (@p_USER_APPROVES IS NULL OR @p_USER_APPROVES="" OR GR.USER_APPROVES LIKE '%'+@p_USER_APPROVES+'%')
AND (
(A.MAKER_ID=@p_USER_LOGIN)
OR (@p_USER_LOGIN = 'baotq')--NGUYENTD 27122023_SECRETKEY: Y/C CỦA ANH BẢO
OR (EXISTS )
OR EXISTS (SELECT 1 FROM CM_TEMPLATE_SHARE WHERE SHARE_USERNAME = @p_USER_LOGIN AND REQ_ID =
A.REQUEST_TEMPLATE_ID)
OR (EXISTS 1 FROM #LIST_ROLE USERROLE WHERE USERROLE.ROLE_ID = 'ADMIN_TO_TRINH' ))
--OR (dbo.FN_CHECK_ROLE_USER(@p_USER_LOGIN,'ADMIN_TO_TRINH')=1)
OR (EXISTS (SELECT top(1) 1 FROM #LIST_ROLE USERROLE WHERE USERROLE.ROLE_ID = 'GDDV'
AND (
EXISTS 1 FROM TL_USER US WHERE US.TLNANME = A.MAKER_ID AND (USERROLE.BRANCH_ID = US.TLSUBBRID OR
USERROLE.DEPT_ID = US.DEP_ID) AND US.TLSUBBRID <> 'DV0001')
OR EXISTS 1 FROM TL_USER US WHERE US.TLNANME = A.MAKER_ID AND USERROLE.DEPT_ID = US.DEP_ID)
)
)
)
)
--NGUYENTD 17032024_SECRETKEY: FIX TÌM KIẾM THEO ĐƠN VỊ TẠO
AND (BR.BRANCH_ID LIKE '%'+ @p_BRANCH_ID + '%' OR @p_BRANCH_ID IS NULL OR @p_BRANCH_ID = "")
AND (A.BRANCH_ID LIKE '%'+ @p_DEP_ID + '%' OR @p_DEP_ID IS NULL OR @p_DEP_ID = "")
ORDER BY A.CREATE_DT DESC
--END NGUYENTD
--END WORKFLOW
-- PAGING END
END
END -- PAGING
DROP TABLE #LIST_ROLE
DROP TABLE #AUTH_STATUS
DROP TABLE #CM_REQUEST_TEMPLATE
DROP TABLE #SHARE
DROP TABLE #APPROVE_GROUP
DROP TABLE #BRANCH

```

#6 - 03/25/2025 04:34 PM - Luc Tran Van

```

;WITH LatestSteps AS (
SELECT
REQ_ID,
MAX AS MaxStepLevel,
STUFF((
SELECT DISTINCT ',' + CAST)
FROM cm_approve_group b
WHERE b.REQ_ID = a.REQ_ID
FOR XML PATH
), 1, 1, '') AS UniqueStepLevels,
COUNT AS DistinctStepCount
FROM cm_approve_group a
WHERE PROCESS_STATUS = 1
AND EXISTS (
SELECT 1
FROM cm_request_template c
WHERE c.REQUEST_TEMPLATE_ID = a.REQ_ID
AND c.AUTH_STATUS = 'A'
)
GROUP BY REQ_ID

```

```

HAVING COUNT >= 2 -- ít nhất 2 STEP_LEVEL khác nhau
)
SELECT
cag.*,
ls.UniqueStepLevels,
ls.DistinctStepCount
FROM cm_approve_group cag
JOIN LatestSteps ls ON cag.REQ_ID = ls.REQ_ID
WHERE cag.PROCESS_STATUS = 1
AND EXISTS (
SELECT 1
FROM cm_request_template a
WHERE a.REQUEST_TEMPLATE_ID = cag.REQ_ID
AND a.AUTH_STATUS = 'A'
)
ORDER BY cag.CREATE_DT DESC;

```

#7 - 03/26/2025 11:15 AM - Luc Tran Van

```

WITH NumberedLogs AS (
SELECT
t.TEMPLATE_REQUEST_ID,
t.GROUP_APPROVE,
ROW_NUMBER() OVER (ORDER BY t.TEMPLATE_REQUEST_ID) AS RowNum
FROM CM_TEMPLATE_APPROVE_LOG t
INNER JOIN (
SELECT TEMPLATE_REQUEST_ID, MAX AS MaxCreateDT
FROM CM_TEMPLATE_APPROVE_LOG
GROUP BY TEMPLATE_REQUEST_ID
) m ON t.TEMPLATE_REQUEST_ID = m.TEMPLATE_REQUEST_ID
AND t.CREATE_DT = m.MaxCreateDT
WHERE t.TEMPLATE_REQUEST_ID IN (
SELECT REQUEST_TEMPLATE_ID
FROM CM_REQUEST_TEMPLATE
WHERE AUTH_STATUS IN ('O','A')
AND t.TEMPLATE_REQUEST_ID = 'CRTEM0000018492'
)
),
LatestLogs AS (
SELECT
TEMPLATE_REQUEST_ID,
GROUP_APPROVE
FROM NumberedLogs
WHERE RowNum BETWEEN 1 AND 50
),
SplitUsernames AS (
-- Bước 1: Split bằng dấu phẩy để lấy từng nhóm
SELECT
LL.TEMPLATE_REQUEST_ID,
GroupSplit.[value] AS GroupValue,
ROW_NUMBER() OVER (PARTITION BY LL.TEMPLATE_REQUEST_ID ORDER BY (SELECT 1)) AS STEP_LEVEL
FROM LatestLogs LL
CROSS APPLY WSISPLIT AS GroupSplit
WHERE GroupSplit.[value] IS NOT NULL AND GroupSplit.[value] <> "
),
FinalSplit AS (
-- Bước 2: Split từng nhóm bằng dấu chấm phẩy để lấy username
SELECT
TEMPLATE_REQUEST_ID,
STEP_LEVEL,
UserSplit.[value] AS USERNAME
FROM SplitUsernames
CROSS APPLY WSISPLIT AS UserSplit
WHERE UserSplit.[value] IS NOT NULL AND UserSplit.[value] <> "
)
SELECT
SLC.*,
AG.*
FROM CM_APPROVE_GROUP AG
INNER JOIN FinalSplit SLC
ON AG.REQ_ID = SLC.TEMPLATE_REQUEST_ID
AND AG.APPROVE_USERNAME = SLC.USERNAME
WHERE AG.STEP_LEVEL <> SLC.STEP_LEVEL AND AG.type = 'APP'

```

SET QUOTED_IDENTIFIER ON

SET ANSI_NULLS ON

GO

ALTER PROCEDURE [dbo].[rpt_PL_REQUEST_FUND_Template]

@p_TR_REQUEST_DOC_ID varchar(15) = NULL

AS

```
-----CHỮ KÝ DMMS-----
DECLARE @t_SIGN_TABLE_DMMS TABLE(
    APPROVE_DT VARCHAR(30), SIGN_NAME NVARCHAR(50), TYPE VARCHAR(15)
)
DECLARE @p_APPROVE_DT_KSV VARCHAR(25), @p_KSV_NAME NVARCHAR(50), @p_ROLE_KSV NVARCHAR(200), --KSV
        @p_APPROVE_DT_GDDV VARCHAR(25), @p_GDDV_NAME NVARCHAR(50), --GDDV
        @p_ROLE_PP NVARCHAR(100) = N'PHÓ PHÒNG HÀNH CHÍNH', @p_APPROVE_DT_PP VARCHAR(25), @p_PP_NAME NVARCHAR(50)
) --KSV/PP.HC

INSERT INTO @t_SIGN_TABLE_DMMS
    -----KSV DMMS-----
    SELECT TOP 1 ISNULL(FORMAT(D.APPROVE_DT, 'dd/MM/yyyy,HH:mm:ss'),'') AS APPROVE_DT,
    C.TLFullName AS SIGN_NAME,
    'KSV' AS TYPE
    FROM PL_REQUEST_PROCESS_CHILD A
    LEFT JOIN PL_REQUEST_PROCESS B ON A.PROCESS_ID = B.ID
    LEFT JOIN TL_USER C ON A.TLNAME = C.TLNANME
    LEFT JOIN PL_PROCESS D ON B.REQ_ID = D.REQ_ID AND A.TLNAME = D.CHECKER_ID
    WHERE B.REQ_ID = @p_TR_REQUEST_DOC_ID
    AND A.TYPE_JOB = 'KS'
    AND A.STATUS_JOB = 'P'
    ORDER BY D.ID DESC

INSERT INTO @t_SIGN_TABLE_DMMS
    -----Trưởng đơn vị DMMS-----
    SELECT
    DISTINCT ISNULL(FORMAT(A.APPROVE_DT, 'dd/MM/yyyy,HH:mm:ss'),'') AS APPROVE_DT,
    B.TLFullName AS SIGN_NAME,
    'GDDV' AS TYPE
    FROM PL_REQUEST_PROCESS A
    LEFT JOIN TL_USER B ON A.CHECKER_ID = B.TLNANME
    WHERE A.REQ_ID = @p_TR_REQUEST_DOC_ID
    AND A.PROCESS_ID = 'DMMS'
    AND A.STATUS = 'P'
    /*===== PP.HC DMMS =====*/
INSERT INTO @t_SIGN_TABLE_DMMS
    SELECT TOP 1 ISNULL(FORMAT(D.APPROVE_DT, 'dd/MM/yyyy,HH:mm:ss'),'') AS APPROVE_DT,
    C.TLFullName AS SIGN_NAME,
    'PP' AS TYPE
    FROM PL_REQUEST_PROCESS_CHILD A
    LEFT JOIN PL_REQUEST_PROCESS B ON A.PROCESS_ID = B.ID
    LEFT JOIN TL_USER C ON A.TLNAME = C.TLNANME
    LEFT JOIN PL_PROCESS D ON B.REQ_ID = D.REQ_ID AND A.TLNAME = D.CHECKER_ID
    WHERE B.REQ_ID = @p_TR_REQUEST_DOC_ID
    AND A.TYPE_JOB = 'KS'
    AND A.STATUS_JOB = 'P'
    AND A.TLNAME = 'vanpt2'
    ORDER BY D.ID DESC

-----KSV DMMS-----
SET @p_ROLE_KSV = ( SELECT TOP 1 IIF(A.TLNAME = 'vanpt2', N'PP.HC MUA SẮM VÀ CUNG ỨNG',N'KIỂM SOÁT VIÊN')
    FROM PL_REQUEST_PROCESS_CHILD A
    LEFT JOIN PL_REQUEST_PROCESS B ON A.PROCESS_ID = B.ID
    LEFT JOIN TL_USER C ON A.TLNAME = C.TLNANME
    LEFT JOIN PL_PROCESS D ON B.REQ_ID = D.REQ_ID AND A.TLNAME = D.CHECKER_ID
    WHERE B.REQ_ID = @p_TR_REQUEST_DOC_ID
    AND A.TYPE_JOB = 'KS'
    AND A.STATUS_JOB = 'P'
    ORDER BY D.ID DESC)

SET @p_APPROVE_DT_KSV = (SELECT A.APPROVE_DT FROM @t_SIGN_TABLE_DMMS A WHERE A.TYPE = 'KSV')
IF (@p_APPROVE_DT_KSV IS NULL OR @p_APPROVE_DT_KSV = '')
BEGIN
    SET @p_APPROVE_DT_KSV = (SELECT A.APPROVE_DT FROM @t_SIGN_TABLE_DMMS A WHERE A.TYPE = 'PP')
END
SET @p_KSV_NAME = (SELECT A.SIGN_NAME FROM @t_SIGN_TABLE_DMMS A WHERE A.TYPE = 'KSV')
```

```

IF (@p_KSV_NAME IS NULL OR @p_KSV_NAME = '')
BEGIN
    SET @p_KSV_NAME = (SELECT A.SIGN_NAME FROM @t_SIGN_TABLE_DMMS A WHERE A.TYPE = 'PP')
END
-----GDDV DMMS-----
SET @p_APPROVE_DT_GDDV = (SELECT A.APPROVE_DT FROM @t_SIGN_TABLE_DMMS A WHERE A.TYPE = 'GDDV')
SET @p_GDDV_NAME = (SELECT A.SIGN_NAME FROM @t_SIGN_TABLE_DMMS A WHERE A.TYPE = 'GDDV')
----- PP DMMS -----
SET @p_APPROVE_DT_PP = (SELECT A.APPROVE_DT FROM @t_SIGN_TABLE_DMMS A WHERE A.TYPE = 'PP')
SET @p_PP_NAME = (SELECT A.SIGN_NAME FROM @t_SIGN_TABLE_DMMS A WHERE A.TYPE = 'PP')

DECLARE @REQ_DT DATETIME, @CREATE_PL_APP DATETIME
SELECT TOP 1 @REQ_DT = REQ_DT, @CREATE_PL_APP = CREATE_DT FROM PL_APPOINT_CONTRACTOR WHERE TR_REQUEST_DOC_
ID = @p_TR_REQUEST_DOC_ID
-----LẤY TÊN NG DUYỆT-----
SELECT      ISNULL(@p_KSV_NAME, '') AS KSV_NAME,
            ISNULL(@p_GDDV_NAME, '') AS DMMS_NAME,
            ISNULL(@p_PP_NAME, '') AS PP_NAME
-----LẤY NGÀY DUYỆT-----
IF (CONVERT (DATE, @REQ_DT, 103) < CONVERT (DATE, @CREATE_PL_APP, 103))
BEGIN
    SELECT CASE WHEN ISNULL(@p_APPROVE_DT_KSV, '') <> '' THEN ISNULL (FORMAT (@REQ_DT, 'dd/MM/yyyy,HH:mm:ss'
), '')
            ELSE '' END AS APPROVE_DT_KSV,
            CASE WHEN ISNULL (@p_APPROVE_DT_GDDV, '') <> '' THEN ISNULL (FORMAT (@REQ_DT, 'dd/MM/yyyy,HH:mm:ss'), '')
)
            ELSE '' END AS APPROVE_DT_GDDV,
            CASE WHEN ISNULL (@p_APPROVE_DT_PP, '') <> '' THEN ISNULL (FORMAT (@REQ_DT, 'dd/MM/yyyy,HH:mm:ss'), '')
            ELSE '' END AS APPROVE_DT_PP
END
ELSE
BEGIN
    SELECT      ISNULL (@p_APPROVE_DT_KSV, '') AS APPROVE_DT_KSV,
            ISNULL (@p_APPROVE_DT_GDDV, '') AS APPROVE_DT_GDDV,
            ISNULL (@p_APPROVE_DT_PP, '') AS APPROVE_DT_PP
END
END

-----CHỮ KÝ GDK hỗ trợ-----
SELECT
    CASE
        WHEN PRP.PROCESS_ID = 'GDK_PYC'
            THEN N'PHÊ DUYỆT'
        END AS TITLE,
    CEL.POS_NAME AS ROLE_GDKHT,
    TU.TLFullName AS GDKHT_NAME,
    ISNULL (FORMAT (PRP.APPROVE_DT, 'dd/MM/yyyy,HH:mm:ss'), '') AS APPROVE_DT_GDKHT
FROM PL_REQUEST_PROCESS PRP
LEFT JOIN TL_USER TU ON PRP.CHECKER_ID = TU.TLNANME
LEFT JOIN CM_EMPLOYEE_LOG CEL ON TU.TLNANME = CEL.USER_DOMAIN
WHERE PRP.REQ_ID = @p_TR_REQUEST_DOC_ID
AND PRP.PROCESS_ID = 'GDK_PYC'
AND PRP.DVDM_ID = 'DM00000000000015' ---PHÒNG HỖ TRỢ
AND PRP.STATUS = 'P'
GO

```

#9 - 04/04/2025 08:50 AM - Luc Tran Van

```

SELECT B.YEAR, A.*
FROM dbo.PL_TRADEDETAIL A
INNER JOIN dbo.PL_MASTER B ON B.PLAN_ID = A.PLAN_ID
WHERE TRADE_ID IN (
SELECT A.TRADE_ID
FROM dbo.PL_TRADEDETAIL A
INNER JOIN dbo.PL_MASTER B ON B.PLAN_ID = A.PLAN_ID
WHERE B.YEAR = '2025'
AND EXISTS (
SELECT 1
FROM dbo.PL_TRADEDETAIL C
WHERE C.PLAN_ID = A.PLAN_ID
AND C.GOODS_ID = A.GOODS_ID
AND C.BRANCH_ID = A.BRANCH_ID
AND C.DEP_ID = A.DEP_ID
GROUP BY C.GOODS_ID, C.BRANCH_ID, C.DEP_ID
HAVING COUNT >= 2
)
)

```

)