



pears.EscalatePopups



Generated schema reference. Regenerate this page from the SQL unload; keep hand-maintained business notes in the narrative namespace.

Original SQL

```
CREATE PROCEDURE "pears"."EscalatePopups"()
/* Application Maintained Function / Procedure - DO NOT EDIT*/
BEGIN
  DECLARE "idum" SMALLINT;
  IF NOT EXISTS(SELECT * FROM "params" WHERE "EscalatePopups" = 1) THEN
    RETURN
  END IF;
  BEGIN
    CREATE variable "uniqueroll" SMALLINT
  exception
    WHEN others THEN SET "idum" = 0
  END;
  FOR "forlab" AS "curs" insensitive cursor FOR
    SELECT "diary"."diaryid" AS "xxxdiaryid","diary"."staffid" AS
"xxxstaffid","staff"."userid" AS "xxxuserid",
      "datediff"("minute",CURRENT TIMESTAMP,"diaryfrom") AS
"xxxpopmins","OriginalDiaryID" AS "xxxOriginalDiaryID","WhoEntered" AS
"xxxWhoEntered" FROM "diary" KEY JOIN "staff"
      WHERE "diary"."popuptime" <= "dateadd"("minute",-5,CURRENT TIMESTAMP)
AND "canescalate" IS NULL AND "diary"."staffid" = any(SELECT "origstaffid"
FROM "popupescalation") FOR READ ONLY do
    IF "xxxpopmins" between -32700 AND 32700 THEN
      INSERT INTO "diary"(
        "diaryid","staffid","personid","vacancyid","employmentid","progressid","plac
ementid","diaryfrom","diaryto",
        "description","notes","durationtype","duration","popupminutes","OriginalDiar
yID","WhoEntered" ) (
        SELECT
        "uniquekey"("string"("diaryid","TargetStaffID")),"targetstaffid","personid",
        "vacancyid","employmentid","progressid","placementid","diaryfrom","diaryto",
          "description","string"('ESCALATED from
',"xxxuserid","char"(13),"char"(10),"notes"),
          "durationtype","duration","xxxpopmins",
          "isnull"("xxxOriginalDiaryID","xxxDiaryID"), //IF this IS the
FIRST escalation xxxOriginalDiaryID will be NULL
```



```
"xxxWhoEntered"
FROM "diary","popupescalation"
WHERE "diary"."diaryid" = "xxxdiaryid"
AND "popupescalation"."origstaffid" = "xxxstaffid"
AND NOT EXISTS(SELECT "DiaryID" FROM "Diary" WHERE "StaffID" =
"TargetStaffID" AND "OriginalDiaryID" = "xxxOriginalDiaryID")); // Prevents
multiple reminders FOR circular escalations
-- clear pop-up time on escalated pop-ups and indicate where escalated
UPDATE "diary" SET "popupminutes" = NULL,"notes" = "string('ESCALATED
to ',
    (SELECT "list"("staff"."userid") FROM "staff" WHERE
"popupescalation"."targetstaffid" = "staff"."staffid"),
    "char"(13),"char"(10),"notes") FROM
    "diary","popupescalation"
WHERE "diaryid" = "xxxdiaryid"
AND "popupescalation"."origstaffid" = "xxxstaffid"
AND NOT EXISTS(SELECT "DiaryID" FROM "Diary" WHERE "StaffID" =
"TargetStaffID" AND "OriginalDiaryID" = "xxxOriginalDiaryID")
ELSE
    UPDATE "diary" SET "popupminutes" = NULL WHERE "diaryid" =
"xxxdiaryid"
END IF
END FOR
END
GO

COMMENT TO PRESERVE FORMAT ON PROCEDURE "pears"."EscalatePopups" IS
{CREATE PROCEDURE EscalatePopups

/* Application Maintained Function / Procedure - DO NOT EDIT*/

()
BEGIN
    DECLARE idum SMALLINT;
    IF NOT EXISTS(SELECT * FROM params WHERE EscalatePopups = 1) THEN
        RETURN
    END IF;
    BEGIN
        CREATE variable uniqueroll SMALLINT
    exception
        WHEN others THEN SET idum=0
    END;
    FOR forlab AS curs insensitive cursor FOR
        SELECT diary.diaryid AS xxxdiaryid,diary.staffid AS
xxxstaffid,staff.userid AS xxxuserid,
        datediff(MINUTE,CURRENT TIMESTAMP,diaryfrom) AS
xxxpopmins,OriginalDiaryID AS xxxOriginalDiaryID,WhoEntered AS xxxWhoEntered
    FROM diary KEY JOIN staff WHERE
```



```
diary.popupptime <= dateadd(MINUTE,-5,CURRENT_TIMESTAMP) AND
canescalate IS NULL AND diary.staffid = any(SELECT origstaffid FROM
popupescalation) FOR READ ONLY do
    IF xxxpopmins between -32700 AND 32700 THEN
        INSERT INTO diary(
diaryid,staffid,personid,vacancyid,employmentid,progressid,placementid,diary
from,diaryto,
description,notes,durationtype,duration,popupminutes,OriginalDiaryID,WhoEnte
red) (
    SELECT
uniquekey(string(diaryid,TargetStaffID)),targetstaffid,personid,vacancyid,em
ploymentid,progressid,placementid,diaryfrom,diaryto,
    description,string('ESCALATED from
',xxxuserid,"char"(13),"char"(10),notes),
    durationtype,duration,xxxpopmins,
    isnull(xxxOriginalDiaryID,xxxDiaryID), //IF this IS the FIRST
escalation xxxOriginalDiaryID will be NULL
    xxxWhoEntered FROM
diary,popupescalation WHERE
diary.diaryid = xxxdiaryid AND
popupescalation.origstaffid = xxxstaffid AND
    NOT EXISTS(SELECT DiaryID FROM Diary WHERE StaffID = TargetStaffID
AND OriginalDiaryID = xxxOriginalDiaryID)); // Prevents multiple reminders
FOR circular escalations
    -- clear pop-up time on escalated pop-ups and indicate where escalated
    UPDATE diary SET popupminutes = NULL,notes = string('ESCALATED to ',
(SELECT list(staff.userid) FROM staff WHERE
popupescalation.targetstaffid = staff.staffid),
"char"(13),"char"(10),notes) FROM
diary,popupescalation WHERE
diaryid = xxxdiaryid AND
popupescalation.origstaffid = xxxstaffid AND
    NOT EXISTS(SELECT DiaryID FROM Diary WHERE StaffID = TargetStaffID
AND OriginalDiaryID = xxxOriginalDiaryID)
    ELSE
        UPDATE diary SET popupminutes = NULL WHERE diaryid = xxxdiaryid
    END IF END FOR
END
}
```

From:
<https://iqxusers.co.uk/iqxhelp/> - iqx

Permanent link:
https://iqxusers.co.uk/iqxhelp/doku.php?id=database:procedures:pears_escalatepopups

Last update: 2026/08/07 19:24

