



pears.TempTimesheetCancel



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

Original SQL

```
CREATE FUNCTION "pears"."TempTimesheetCancel"(  
    /* Application Maintained Function / Procedure - DO NOT EDIT*/  
    IN "tscancelid" CHAR(20),IN "tsid" CHAR(20),IN "shiftcancreasonid"  
    CHAR(1),IN "irecreate" SMALLINT,IN @CorrectionType CHAR(20) DEFAULT 'BOTH' )  
    RETURNS CHAR(20)  
    BEGIN atomic  
        DECLARE "newid" CHAR(20);  
        DECLARE "pid" CHAR(20);  
        DECLARE "stdate" DATE;  
        DECLARE "contrasuffix" CHAR(10);  
        DECLARE "replacesuffix" CHAR(10);  
        -- Temporary table to link the old shifts to the new  
        DECLARE LOCAL TEMPORARY TABLE "localshifts"(  
            "oldshiftid" CHAR(20) NULL,  
            "oldtemplateid" CHAR(20) NULL,  
            "oldstaffid" CHAR(20) NULL,  
            "newshiftid" CHAR(20) NULL,  
        ) ON commit DELETE ROWS;IF "trim"("shiftcancreasonid") = '' THEN  
            SET "shiftcancreasonid" = NULL  
        END IF; -- Avoid ref integ. problems  
        SELECT  
            "isnull"("WPKMaintainGetSwitchValue"('TSCONTRASUFFIX',NULL,'L'),'(C)') INTO  
            "contrasuffix";  
        SELECT  
            "isnull"("WPKMaintainGetSwitchValue"('TSREPLACESUFFIX',NULL,'L'),'(A)') INTO  
            "replacesuffix";  
        SET "newid" = "uniquekey"("tsid");  
        -- Temporary table to link the old shifts to the new  
        UPDATE "temptimesheet" SET "timesheetcancelreasonid" = IF "tscancelid" >  
            '' THEN "tscancelid" ELSE NULL endif WHERE "temptimesheetid" = "tsid";  
        -- newid is the id for both the contra timesheet and the replacement prov  
        timesheet  
        -- First create the contra timesheet (serialnumber suffixed with (C) or  
        contrasuffix)  
        INSERT INTO "temptimesheet"(  

```



```
"temptimesheetid","placementid","serialnumber","period","periodlength","payr
ollidentifier","analysiscode","correctionflag","personid","tempdeskid","payr
ollnumber","theirref","description",
"currency","startdate","enddate","secondaryagencyid","staffid","taxmethod","
holidaypaystatus","TSBarcode","NotTaxable","billed","paid","AWRWasInvolved",
"timesheetcancelreasonid",
    "invoiceprefix","invoiceemail","invoiceaddress" )
SELECT
"newid","placementid","string"("serialnumber","contrasuffix"),"period","peri
odlength","payrollidentifier","analysiscode",1,"personid","tempdeskid","payr
ollnumber","theirref","string"('CANCELLED-',"description"),
"currency","startdate","enddate","secondaryagencyid","userstaffid","taxmetho
d","holidaypaystatus","TSBarcode","NotTaxable",
    IF(@CorrectionType = 'PAY' OR @CorrectionType = 'INV') THEN 1 ELSE
endif,IF @CorrectionType = 'CHARGE' THEN 1 ELSE IF @CorrectionType = 'INV'
THEN "paid" ELSE 0 endif endif,
"AWRWasInvolved","timesheetcancelreasonid","invoiceprefix","invoiceemail","i
nvoiceaddress" FROM "temptimesheet" WHERE "temptimesheetid" = "tsid";
INSERT INTO "temptimesheetline"(
"TempTimeSheetLineID","TempTimeSheetID","TempPayBandID","LineNumber","UnitsP
aid","UnitsCharged","PayRate","ChargeRate","Description","TempShiftID","Shif
tTime","Grade","AWRJobmasterID" )
SELECT
"uniquekey"("TempTimeSheetLineID"),"newid","TempPayBandID","LineNumber",0-
"UnitsPaid",0-
"UnitsCharged","PayRate","ChargeRate","Description","TempShiftID","ShiftTime
","Grade","AWRJobmasterID"
    FROM "temptimesheetline" WHERE "temptimesheetid" = "tsid";
-- create pay_emp
SELECT "personid","startdate" INTO "pid","stdate" FROM "temptimesheet"
WHERE "temptimesheetid" = "newid";
CALL "pay_checkemployment"("pid","stdate");
-- update ts
UPDATE "temptimesheet" SET "holidaypaystatus" =
"tempholidaycalc"("p"."personid","truncnum"("period"/100,0),"remainder"("per
iod",100),-1),
    "taxmethod" = "e"."taxmethod","pay_employmentid" = (SELECT FIRST
"pay_employmentid" FROM "pay_employment" WHERE "enddate" IS NULL AND
"personid" = "temptimesheet"."personid"),
    "payrollnumber" = "p"."payrollnumber" FROM
    "temptimesheet" KEY JOIN "person" AS "p" KEY JOIN "pay_employee" AS "e"
WHERE "temptimesheetid" = "newid";
IF EXISTS(SELECT * FROM "person" WHERE "personid" = "pid" AND
"isnull"("payrollnumber",'') = '') THEN
    UPDATE "temptimesheet" SET "payrollnumber" = 'NEW' WHERE
"temptimesheetid" = "newid";
    UPDATE "person" SET "payrollnumber" = 'NEW' WHERE "personid" = "pid"
END IF;
```



```
IF(SELECT FIRST "payrollcoidfromtemp" FROM "params") = 1 THEN
  UPDATE "temptimesheet" SET "payrollidentifier" = (SELECT FIRST
"payrollidentifier" FROM "person" WHERE "personid" = "pid") WHERE
"temptimesheetid" = "newid"
END IF;
-- call cancel
IF "irecreate" = 1 THEN
  -- Populate the local temp table
  INSERT INTO "localshifts"(
"oldshiftid","oldstaffid","newshiftid","oldtemplateid" ) SELECT
"tempshiftid","staffid","uniquekey"("tempshiftid"),"tempshifttemplateid"
FROM "tempshift" WHERE "temptimesheetid" = "tsid";
  -- Create the replacement shifts. Only tricky thing is cancelled shifts
which were billed or paid. We attempt to recreate CBill and CPay values from
the cancel reason
  INSERT INTO "tempshift"(
"Staffid","TempShiftID","VacancyID","PersonID","PlacementID","ShiftDate","Ti
meFrom","TimeTo","State","Note","TempShiftPlanID","BreakMinutes","Essentials
killGradeID","CancelReason",
"WhenCancelled","CBill","CPay","CRefill","ReferenceCode","ClientConfirmed","
TempConfirmed","ConfirmedWith","TempShiftTypeID","AnalysisCode","WhoCancelle
d","RecoveryHours","tempshifttemplateid" )
  SELECT
"oldstaffid","l"."newshiftid","VacancyID","PersonID","PlacementID","ShiftDat
e","TimeFrom","TimeTo",(IF "s"."State" = 'W' THEN 'B' ELSE "s"."State"
endif),
"Note","TempShiftPlanID","BreakMinutes","EssentialSkillGradeID","CancelReaso
n",
"WhenCancelled","c"."Bill","c"."Pay","CRefill","ReferenceCode",1,1,"Confirme
dWith","TempShiftTypeID","AnalysisCode","WhoCancelled","RecoveryHours","oldt
emplateid"
  FROM "localshifts" AS "l" JOIN "tempshift" AS "s" ON
"l"."oldshiftid" = "s"."tempshiftid" KEY LEFT OUTER JOIN
"tempshiftcancelreason" AS "c";
  -- Create the replacement provisional (serialnumber suffixed with (A) or
replacesuffix)
  INSERT INTO "tempprovtimesheet"(
"TempProvTimeSheetID","PlacementID","PersonID","VacancyID","TempDeskID","Per
iod","PeriodLength","CorrectionFlag","Description","TheirRef","SerialNumber"
,"ImageBlobID","ScanPathFilename","Orientation","ImageApproved" )
  SELECT "newid","placementid","personid",(SELECT "vacancyid" FROM
"placement" WHERE "placementid" =
"temptimesheet"."placementid"),"tempdeskid","period","periodlength",1,"strin
g"('AMENDED-', "description"),
"theirref","string"("serialnumber","replacesuffix"),"ImageBlobID","ScanPathF
ilename","Orientation","ImageApproved" FROM "temptimesheet" WHERE
"temptimesheetid" = "tsid";
  INSERT INTO "tempprovtimesheetline"(
```



```
"TempProvTimeSheetLineID", "TempProvTimeSheetID", "TempPayBandID", "LineNumber",
"UnitsPaid", "UnitsCharged", "PayRate", "ChargeRate", "Description", "tempshiftid",
    "ShiftTime", "Grade" )
SELECT
"uniquekey" ("TempTimeSheetLineID"), "newid", "TempPayBandID", "LineNumber", "UnitsPaid",
"UnitsCharged", "PayRate", "ChargeRate", "Description",
    (SELECT FIRST "newshiftid" FROM "localshifts" WHERE "oldshiftid" =
"temptimesheetline"."tempshiftid"),
    "ShiftTime", "Grade"
    FROM "temptimesheetline" WHERE "temptimesheetid" = "tsid";
-- Link the new shifts to it
INSERT INTO "tempprovtimesheetshift" (
"tempprovtimesheetid", "tempshiftid" ) SELECT "newid", "newshiftid" FROM
"localshifts"
END IF;
-- Cancel the old shifts (unless already cancelled), using the designated
cancel reason
UPDATE "tempshift" SET "state" = 'C', "cancelreason" =
"shiftcancelreasonid", "whencancelled" = CURRENT_TIMESTAMP, "whocancelled" =
(SELECT FIRST "staffid" FROM "temptimesheet" WHERE "temptimesheetid" =
"newid")
WHERE "temptimesheetid" = "tsid" AND "state" <> 'C';
-- Record the cancellation in the TempTimesheetCancel table
INSERT INTO "TempTimesheetCancellation" (
"TempTimeSheetCancellationID", "OriginalTempTimesheetID", "PayCorrected", "ChargeCorrected",
"ReplacementProvTimesheetSerialNumber", "ContraTempTimeSheetID",
"WhoCancelled", "TimesheetCancelReasonID" ) VALUES
( "uniquekey" ("tsid"), "tsid", IF @CorrectionType IN( 'BOTH', 'PAY' ) THEN
1 ELSE 0 endif, IF @CorrectionType IN( 'BOTH', 'CHARGE' ) THEN 1 ELSE 0 endif,
IF "irecreate" = 1 THEN (SELECT "string" ("serialnumber", "replacesuffix")
FROM "temptimesheet" WHERE "temptimesheetid" = "tsid") ELSE NULL
endif, "newid", "UserStaffID", IF "tscancelid" > '' THEN "tscancelid" ELSE NULL
endif ) ;
-- If expense benefit in operation, cancel out any EB timesheets or equiv.
Actual content of procedure will vary depending on payroll system etc
IF (SELECT "ExpenseBenefit" FROM "TempDesk" KEY JOIN "Vacancy" KEY JOIN
"Placement" KEY JOIN "TempTimeSheet" WHERE "TempTimeSheetID" = "tsid") = 1
THEN
    CALL "CancelExpenseBenefitTimesheet" ("tsid", "newid")
END IF;
CALL "temptimesheetcomplete" ("newid");
-- Return the ID of the contra and replacement prov timesheets
RETURN "newid"
END
```



From:

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

Permanent link:

https://iqxusers.co.uk/iqxhelp/doku.php?id=database:functions:pears_temptimesheetcancel

Last update: **2026/08/07 19:24**

