



pears.AWRWeeks



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

Original SQL

```
CREATE FUNCTION "pears"."AWRWeeks"(  
    /* Application Maintained Function / Procedure - DO NOT EDIT*/  
    IN @JobMasterID CHAR(20), IN @WeekNo INTEGER, IN @VacID CHAR(20), IN @PersID  
    CHAR(20), IN @PlacID CHAR(20), IN @ShiftID CHAR(20) )  
    RETURNS INTEGER  
    BEGIN  
        DECLARE @AWRStat INTEGER;  
        DECLARE @AWRLink CHAR(20);  
        DECLARE @PersonID CHAR(20);  
        DECLARE @DirectlyEmployed SMALLINT;  
        DECLARE @AWRStartPeriod INTEGER;  
        DECLARE @WkCount INTEGER;  
        DECLARE @GapCount SMALLINT;  
        DECLARE @FunctionTimeStamp "datetime";  
        DECLARE @LastPeriod INTEGER;  
        DECLARE @Override SMALLINT;  
        DECLARE @OverrideDate DATE;  
        DECLARE @ThisPeriodGap SMALLINT;  
        DECLARE @FirstOne SMALLINT;  
        -- parameters @VacID, @PersID, @PlacID, @ShiftID not currently used  
        -- check override AWR start date not set (mainly a testing facility)  
        IF "varexists"('AWRStartPeriod') = 1 THEN  
            SET @AWRStartPeriod = "AWRStartPeriod"  
        ELSE SET @AWRStartPeriod =  
"isnull"("weekcontaining"("dateadd"("year",-5,CURRENT DATE)),(SELECT  
"min"("period") FROM "payrollcalendar"))  
        END IF;  
        SET @WkCount = 0;  
        -- check role qualified  
        IF "nullif"(@JobMasterID,'') IS NOT NULL THEN  
            SELECT  
"AWRStatus","AWRLinkCode","personid","isnull"("OverrideSystem",1),"Overrided  
ate",  
            (IF EXISTS(SELECT * FROM "person" AS "p" WHERE "p"."personid" =  
"AWRJobMaster"."Personid" AND "isnull"("directlyemployed",0) >= 1) THEN 1
```



```
ELSE 0 endif)
    INTO
@AWRStat,@AWRLink,@PersonID,@Override,@OverrideDate,@DirectlyEmployed
    FROM "AWRJobMaster" WHERE "AWRJobMasterID" = @JobMasterID;
-- finish if role not qualified or Directly Employed
IF @AWRStat = 3 OR @DirectlyEmployed = 1 THEN
    RETURN 0
END IF;
-- look for override
IF @OverrideDate IS NOT NULL THEN // nested ifs FOR performance -
prevents FUNCTION CALL IF NOT necessary
    IF @WeekNo >= "weekcontaining"(@OverrideDate) THEN
        IF @Override = 3 THEN RETURN 0
        ELSEIF @Override = 2 THEN RETURN 15
        END IF END IF END IF;
-- count weeks since 6 week gap add 1 for continue 0 for pause stop at
15
SET @FunctionTimeStamp = CURRENT TIMESTAMP;
SET @WkCount = 0;
SET @GapCount = 0;
SET @LastPeriod = @WeekNo;
SET @ThisPeriodGap = 0;
SET @FirstOne = 1;
-- populate temporary table
INSERT INTO "AWRWeekFind"(
"AWRJobMasterID","DataGenerated","ThisPeriod","Qual" )
    SELECT @JobMasterID,@FunctionTimeStamp,"period","isnull"("qualtype",0)
    FROM "AWRWeeklyDetail"
    WHERE "AWRJobmasterid" = any(SELECT "AWRJobMasterID" FROM
"AWRJobMaster" WHERE "AWRLinkCode" = @AWRLink)
    AND("disregard" = 0 OR "disregard" IS NULL) AND "period" BETWEEN
@AWRStartPeriod AND @WeekNo UNION ALL
    SELECT @JobMasterID,@FunctionTimeStamp,"period","isnull"("qualtype",0)
    FROM "AWRWeeklyDetail"
    WHERE "personid" = @PersonID AND("disregard" = 0 OR "disregard" IS
NULL) AND "period" BETWEEN @AWRStartPeriod AND @WeekNo UNION ALL
    SELECT @JobMasterID,@FunctionTimeStamp,"period","isnull"("qualtype",0)
    FROM "AWRWeeklyDetail"
    WHERE "companyid" = any(SELECT "companyid" FROM "AWRJobMaster" KEY
JOIN "vacancy" KEY JOIN "employment" WHERE "AWRLinkCode" = @AWRLink)
    AND("disregard" = 0 OR "disregard" IS NULL) AND "period" BETWEEN
@AWRStartPeriod AND @WeekNo;
-- now check results
FOR "forlab" AS "curs" no scroll cursor FOR
    SELECT "ThisPeriod","Qual"
    FROM "AWRWeekFind" WHERE "AWRJobMasterID" = @JobMasterID AND
"DataGenerated" = @FunctionTimeStamp
    ORDER BY "ThisPeriod" DESC,"Qual" DESC FOR READ ONLY
```



```
do -- check if we have moved on to a new period, if not we can skip to
next record
  IF "ThisPeriod" <> @LastPeriod OR @FirstOne = 1 THEN
    -- Work out the gap since the last record
    IF "truncnum"(@LastPeriod,-2)/100 = "truncnum"("ThisPeriod",-2)/100
THEN //same YEAR
    SET @ThisPeriodGap = @LastPeriod-"ThisPeriod"-1
  ELSE //different YEAR
    SET @ThisPeriodGap =
("datediff"("day","weekmonthenddate"("ThisPeriod",'W'),'weekmonthenddate'(@L
astPeriod,'W'))/7)-1
  END IF;
  -- check for 6 week gap since last record and finish if exceeded
  IF @FirstOne = 1 THEN
    SET @ThisPeriodGap = @ThisPeriodGap+1;
    SET @FirstOne = 0
  END IF;
  IF(@GapCount+@ThisPeriodGap >= 6) THEN
    DELETE FROM "AWRWeekFind" WHERE "DataGenerated" =
@FunctionTimeStamp;
    RETURN @WkCount
  END IF;
  -- check the record status
  IF "Qual" = 2 THEN //timesheet, other agency WORK OR other continue
record found
    SET @WkCount = @WkCount+1;
    SET @GapCount = 0
  ELSEIF "Qual" = 1 THEN //pause record found
    SET @GapCount = @GapCount+@ThisPeriodGap
  END IF;
  -- check for 6 week gap and finish if exceeded
  IF @GapCount >= 6 THEN
    DELETE FROM "AWRWeekFind" WHERE "DataGenerated" =
@FunctionTimeStamp;
    RETURN @WkCount
  END IF;
  -- stop counting at 15 weeks since qualification
  IF @WkCount >= 15 THEN
    DELETE FROM "AWRWeekFind" WHERE "DataGenerated" =
@FunctionTimeStamp;
    RETURN @WkCount
  END IF;
  -- set the @lastPeriod ready for moving to next row
  SET @LastPeriod = "ThisPeriod"
END IF END FOR END IF;
DELETE FROM "AWRWeekFind" WHERE "DataGenerated" = @FunctionTimeStamp;
RETURN @WkCount
```



END

From:

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

Permanent link:

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

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

