



pears.PopulatePayrollCalendar



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

Original SQL

```
CREATE PROCEDURE "pears"."PopulatePayrollCalendar"()
/* Application Maintained Function / Procedure - DO NOT EDIT*/
BEGIN
    -- Clears out and then re-populates the Payroll Calendar table after any
    changes
    -- Clear Out Existing Records
    DELETE FROM "PayrollCalendar";
    CALL "sa_reset_identity"('PayrollCalendar','pears',0);
    -- Insert All Defined Dates
    INSERT INTO "PayrollCalendar"( "Year","TheDate","Week" )
        SELECT "YearNumber",
            "TheDate",
            "Week"
        FROM(SELECT "YearNumber",
            "date"("dateadd"("day",-6,"PayrollYear"."Week1Friday")) AS
"StartDate",
            "date"("dateadd"("day",-1,("dateadd"("week","PayrollYear"."NumberOfWeeks","S
tartDate")))) AS "EndDate",
            "date"("dateadd"("day","row_num","StartDate")) AS
"TheDate",("row_num"/7)+1 AS "Week"
            FROM "PayrollYear" CROSS JOIN "sa_rowgenerator"(0,1000)
            WHERE "TheDate" BETWEEN "StartDate" AND "EndDate") AS
"DefinedDates"
            ORDER BY "TheDate" ASC;
    -- Remove Duplicates
    DELETE FROM
        "PayrollCalendar"
        WHERE "PayrollCalendarID" = any(SELECT "a"."PayrollCalendarID" FROM
"PayrollCalendar" AS "a" JOIN "PayrollCalendar" AS "b" ON "a"."TheDate" =
"b"."TheDate" AND "a"."Year" < "b"."Year");
    -- Fill Gaps
    INSERT INTO "PayrollCalendar"( "TheDate","Year","Week" )
        SELECT "Day",
            (SELECT "max"("Year") FROM "PayrollCalendar" WHERE "TheDate" < "Day")
AS "Year",
```



```
(SELECT "max"("Week") FROM "PayrollCalendar" WHERE "TheDate" < "Day")
AS "Week"
FROM "PayrollCalendar"
RIGHT OUTER JOIN(SELECT "date"("dateadd"("day","row_num",(SELECT
"min"("TheDate") FROM "PayrollCalendar")))) AS "Day"
FROM "sa_rowgenerator"(0,(SELECT
"datediff"("day","min"("TheDate"),"max"("TheDate")) FROM
"PayrollCalendar")))) AS "AllDates"
ON "day" = "TheDate" WHERE "TheDate" IS NULL;
-- Populate Other Fields
UPDATE "PayrollCalendar" AS "z"
SET "z"."Period" = ("z"."Year"*100)+"z"."Week",
"WeekBeginning" = (SELECT "min"("a"."TheDate") FROM "PayrollCalendar" AS
"a" WHERE "a"."Year" = "z"."Year" AND "a"."Week" = "z"."Week"),
"WeekEnding" = (SELECT "max"("a"."TheDate") FROM "PayrollCalendar" AS
"a" WHERE "a"."Year" = "z"."Year" AND "a"."Week" = "z"."Week");
-- Populate PayrollCalendarWeek table
DELETE FROM "PayrollCalendarWeek";
CALL "sa_reset_identity"('PayrollCalendarWeek','pears',0);
INSERT INTO "PayrollCalendarWeek"(
"Period","Year","Week","WeekBeginning","WeekEnding" )
SELECT DISTINCT "Period","Year","Week","WeekBeginning","WeekEnding" FROM
"PayrollCalendar" ORDER BY "period" ASC
END
GO

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

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

()
BEGIN
-- Clears out and then re-populates the Payroll Calendar table after any
changes
-- Clear Out Existing Records
DELETE FROM PayrollCalendar;
CALL sa_reset_identity('PayrollCalendar','pears',0);
-- Insert All Defined Dates
INSERT INTO PayrollCalendar(YEAR, TheDate, Week)
SELECT YearNumber,
TheDate,
Week
FROM (
SELECT
YearNumber,
DATE(dateadd(DAY,-6,PayrollYear.Week1Friday)) AS StartDate,
DATE(dateadd(DAY,-1,(dateadd(week,PayrollYear.NumberOfWeeks,StartDate))))AS
```



```
EndDate,
    DATE(dateadd(DAY,row_num,StartDate)) AS TheDate, (row_num/7)+1 AS Week
FROM PayrollYear CROSS JOIN sa_rowgenerator(0,1000)
WHERE TheDate BETWEEN StartDate AND EndDate
) DefinedDates
ORDER BY TheDate;
-- Remove Duplicates
DELETE FROM
    PayrollCalendar
WHERE
    PayrollCalendarID IN (SELECT a.PayrollCalendarID FROM PayrollCalendar a
JOIN PayrollCalendar b ON a.TheDate = b.TheDate AND a.Year < b.Year);
-- Fill Gaps
INSERT INTO PayrollCalendar(TheDate,YEAR,Week)
    SELECT
        DAY,
        (SELECT MAX(YEAR) FROM PayrollCalendar WHERE TheDate < DAY) AS
YEAR,
        (SELECT MAX(Week) FROM PayrollCalendar WHERE TheDate < DAY) AS
Week
    FROM
        PayrollCalendar
    RIGHT OUTER JOIN
        (SELECT
            DATE(dateadd(DAY,row_num,(SELECT MIN(TheDate) FROM
PayrollCalendar))) DAY
            FROM sa_rowgenerator(0,(SELECT
datediff(DAY,MIN(TheDate),MAX(TheDate)) FROM PayrollCalendar))) AS AllDates
        ON DAY = TheDate WHERE TheDate IS NULL;
-- Populate Other Fields
UPDATE PayrollCalendar z
    SET z.Period = (z.Year*100)+z.Week,
        WeekBeginning = (SELECT MIN(a.TheDate) FROM PayrollCalendar a WHERE
a.Year=z.Year AND a.Week=z.Week),
        WeekEnding = (SELECT MAX(a.TheDate) FROM PayrollCalendar a WHERE
a.Year=z.Year AND a.Week=z.Week);
-- Populate PayrollCalendarWeek table
DELETE FROM PayrollCalendarWeek;
CALL sa_reset_identity('PayrollCalendarWeek','pears',0);
INSERT INTO PayrollCalendarWeek(Period, YEAR, Week, WeekBeginning,
WeekEnding)
    SELECT DISTINCT Period, YEAR, Week, WeekBeginning, WeekEnding FROM
PayrollCalendar ORDER BY period;
END
}
```



From:

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

Permanent link:

https://iqxusers.co.uk/iqxhelp/doku.php?id=database:procedures:pears_populatepayrollcalendar

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

