



pears.NetVacancyConfirm



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

Original SQL

```
CREATE PROCEDURE "pears"."NetVacancyConfirm"( IN "pWebUserID" CHAR(20),IN
"pvacancyid" CHAR(20),IN "pshiftplanid" CHAR(20) )
RESULT( "pResult" CHAR(250) )
BEGIN
    DECLARE "spopup" CHAR(100);
    DECLARE LOCAL TEMPORARY TABLE "IDs"(
        "ID" CHAR(20) NULL,
    ) NOT transactional;
    INSERT INTO "IDs"( "ID" )
        SELECT DISTINCT "e"."companyid" FROM "employment" AS "e" KEY JOIN
"iqxnetuserlink" AS "i" WHERE "i"."iqxnetuserid" = "pWebUserID";
    IF NOT "pvacancyid" = any(SELECT "v"."vacancyid" FROM "vacancy" AS "v" KEY
JOIN "employment" AS "e" JOIN "IDs" ON "e"."CompanyID" = "IDs"."ID") THEN
        SELECT '99:~Permission denied';
        RETURN
    END IF;
    IF "IQXNetHasPermission"("pWebUserID",'AUTHORISEJOBS') = 0 THEN
        SELECT '99:~Permission denied';
        RETURN
    END IF;
    SET "pshiftplanid" = "nullif"("trim"("pshiftplanid"), '');
    IF "pshiftplanid" IS NULL THEN
        UPDATE "vacancy" SET "status" = 'C' WHERE "status" = ']' AND "vacancyid"
= "pvacancyid";
        UPDATE "vacancy" KEY JOIN "employment" JOIN "params" ON
"params"."secondarycascade" = 1
        SET "vacancy"."CascadeDateTime" =
"GetCascadeTime"("employment"."CompanyID", "vacancy"."VacancyID", CURRENT
TIMESTAMP, "vacancy"."startdate", CURRENT TIMESTAMP, 0)
        WHERE "vacancy"."vacancyid" = "pvacancyid"
    ELSE
        DELETE FROM "tempshift" WHERE "tempshiftplanid" = "pshiftplanid" AND
"vacancyid" = "pvacancyid" AND "state" = 'C' AND "cancelreason" = '~';
        UPDATE "TempShiftPlan" AS "p" KEY JOIN "vacancy" KEY JOIN "employment"
JOIN "params" ON "params"."secondarycascade" = 1
```



```
        SET "p"."CascadeDateTime" =
"GetCascadeTime"("employment"."CompanyID", "vacancy"."VacancyID", CURRENT
TIMESTAMP, "p"."ShiftDate" + "p"."TimeFrom", CURRENT_TIMESTAMP, 0)
        WHERE "p"."TempShiftPlanID" = "pshiftplanid"
    END IF;
    IF @@rowcount = 0 THEN
        SELECT '1:~Confirmation failed';
        RETURN
    END IF;
    IF "pshiftplanid" IS NOT NULL THEN
        SET "spopup" = 'New shift requirement confirmed ' + "dateformat"((SELECT
FIRST "shiftdate" FROM "tempshiftplan" WHERE "tempshiftplanid" =
"pshiftplanid"), 'Ddd dd/mm/yyyy')
    ELSE
        SET "spopup" = 'New vacancy requirement confirmed'
    END IF;
    CALL
"IQXNetPopup"("pWebUserID", NULL, NULL, "pvacancyid", NULL, NULL, 'ADVVACANCY', "sp
opup", NULL);
    SELECT '0:~Success'
END
GO

COMMENT TO PRESERVE FORMAT ON PROCEDURE "pears"."NetVacancyConfirm" IS
{CREATE PROCEDURE pears."NetVacancyConfirm"( IN "pWebUserID" CHAR(20), IN
"pvacancyid" CHAR(20), IN "pshiftplanid" CHAR(20) )
RESULT( "pResult" CHAR(250) )
BEGIN
    DECLARE "spopup" CHAR(100);
    DECLARE LOCAL TEMPORARY TABLE "IDs"(
        "ID" CHAR(20) NULL,
    ) NOT transactional;
    INSERT INTO "IDs"("ID" )
        SELECT DISTINCT "e"."companyid" FROM "employment" AS "e" KEY JOIN
"iqxnetuserlink" AS "i" WHERE "i"."iqxnetuserid" = "pWebUserID";
    IF NOT "pvacancyid" = any(SELECT "v"."vacancyid" FROM "vacancy" AS "v" KEY
JOIN "employment" AS "e" JOIN "IDs" ON "e"."CompanyID" = "IDs"."ID") THEN
        SELECT '99:~Permission denied';
        RETURN
    END IF;
    IF "IQXNetHasPermission"("pWebUserID", 'AUTHORISEJOBS') = 0 THEN
        SELECT '99:~Permission denied';
        RETURN
    END IF;
    SET "pshiftplanid" = "nullif"("trim"("pshiftplanid"), '');
    IF "pshiftplanid" IS NULL THEN
        UPDATE "vacancy" SET "status" = 'C' WHERE "status" = ']' AND "vacancyid"
= "pvacancyid";
```



```
UPDATE "vacancy" KEY JOIN "employment" JOIN "params" ON
"params"."secondarycascade" = 1
SET "vacancy"."CascadeDateTime" =
"GetCascadeTime"("employment"."CompanyID", "vacancy"."VacancyID", CURRENT
TIMESTAMP, "vacancy"."startdate", CURRENT TIMESTAMP, 0)
WHERE "vacancy"."vacancyid" = "pvacancyid"
ELSE
DELETE FROM "tempshift" WHERE "tempshiftplanid" = "pshiftplanid" AND
"vacancyid" = "pvacancyid" AND "state" = 'C' AND "cancelreason" = '~';
UPDATE "TempShiftPlan" AS "p" KEY JOIN "vacancy" KEY JOIN "employment"
JOIN "params" ON "params"."secondarycascade" = 1
SET "p"."CascadeDateTime" =
"GetCascadeTime"("employment"."CompanyID", "vacancy"."VacancyID", CURRENT
TIMESTAMP, "p"."ShiftDate"+"p"."TimeFrom", CURRENT TIMESTAMP, 0)
WHERE "p"."TempShiftPlanID" = "pshiftplanid"
END IF;
IF @@rowcount = 0 THEN
SELECT '1:~Confirmation failed';
RETURN
END IF;
IF "pshiftplanid" IS NOT NULL THEN
SET "spopup" = 'New shift requirement confirmed '+ "dateformat"((SELECT
FIRST "shiftdate" FROM "tempshiftplan" WHERE "tempshiftplanid" =
"pshiftplanid"), 'Ddd dd/mm/yyyy')
ELSE
SET "spopup" = 'New vacancy requirement confirmed'
END IF;
CALL
"IQXNetPopup"("pWebUserID", NULL, NULL, "pvacancyid", NULL, NULL, 'ADDVACANCY', "sp
opup", NULL);
SELECT '0:~Success'
END
}
```

From:

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

Permanent link:

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

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

