



pears.NetClientRequirementVerifiedCandidates



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

Original SQL

```
CREATE PROCEDURE "pears"."NetClientRequirementVerifiedCandidates"( IN
  "pWebUserID" CHAR(20),IN "pRowID" CHAR(20),IN "IsShift" CHAR(1) )
RESULT( "PersonName" CHAR(60),"PersonID" CHAR(20),"PersonStatus"
  CHAR(20),"StartDate" CHAR(20),"LeaveDate" CHAR(20),"CurrentLeave" DATE )
// IQXWeb
BEGIN
  DECLARE "currentStates" CHAR(100);
  DECLARE LOCAL TEMPORARY TABLE "Cos"(
    "CompanyID" CHAR(20) NULL,
  ) NOT transactional;
  INSERT INTO "Cos"( "CompanyID" )
    SELECT "companyid" FROM "employment" KEY JOIN "iqxnetuserlink" WHERE
    "iqxnetuserid" = "pWebUserID";
  SET "currentStates" = (SELECT FIRST "isnull"("personcurrentstates",'C')
  FROM "params" ORDER BY "dbid" ASC);
  IF "IsShift" = '1' THEN
    SELECT "pe"."name",
      "pe"."personid",
      "st"."name" AS "PersonStatus",
      (SELECT "min"("shiftdate") FROM "tempshift" WHERE "vacancyid" =
      "v"."vacancyid" AND "personid" = "pe"."personid" AND "state" <> 'C') AS
      "StartDate",
      (SELECT "max"("shiftdate") FROM "tempshift" WHERE "vacancyid" =
      "v"."vacancyid" AND "personid" = "pe"."personid" AND "state" <> 'C') AS
      "LeaveDate",
      "LeaveDate" AS "currentleave"
    FROM "tempshiftplan" AS "s"
      KEY JOIN "vacancy" AS "v"
      KEY JOIN "employment" AS "e"
      , "person" AS "pe"
      JOIN "status" AS "st" ON "st"."status" = "pe"."status" AND
      "st"."type" = 'P'
    WHERE "e"."companyid" = any(SELECT "companyid" FROM "cos")
```



```
    AND "s"."tempshiftplanid" = "pRowID"
    AND "pe"."personid" = any(SELECT "personid" FROM "tempshift" WHERE
"vacancyid" = "v"."vacancyid" AND "state" <> 'C')
    AND "locate"("currentStates","pe"."status") > 0
    AND
"EmployeeAcceptable"("pe"."personid","e"."companyid",1,"v"."vacancyid",0) IS
NULL
    AND
"isnull"("personavailableforplan"("pe"."personid","s"."ShiftDate","s"."TimeF
rom","s"."TimeTo","s"."RecoveryHours","s"."minutes",0,"v"."departmentid","s"
"."essentialskill","s"."essentialskillgradeid","s"."essentialskillchoicelist"
,NULL),'A') LIKE '[AM]%'
    ORDER BY "currentleave" DESC,
    "pe"."KeyName" ASC
ELSE
    SELECT "pe"."name",
    "pe"."personid",
    "st"."name" AS "PersonStatus",
    (SELECT "max"("startdate") FROM "employment" WHERE "personid" =
"pe"."personid") AS "StartDate",
    (SELECT "max"("leavedate") FROM "employment" WHERE "personid" =
"pe"."personid") AS "LeaveDate",
    "isnull"("e"."leavedate",CURRENT DATE) AS "currentleave"
    FROM "vacancy" AS "v"
    KEY JOIN "employment" AS "e"
    , "person" AS "pe"
    JOIN "status" AS "st" ON "st"."status" = "pe"."status" AND
"st"."type" = 'P'
    WHERE "e"."companyid" = any(SELECT "companyid" FROM "cos")
    AND "v"."vacancyid" = "pRowID"
    AND "pe"."personid" = any(SELECT "emp"."personid" FROM "employment" AS
"emp" KEY JOIN "placement" AS "pl" WHERE "pl"."vacancyid" = "v"."vacancyid")
    AND "locate"("currentStates","pe"."status") > 0
    AND
"EmployeeAcceptable"("pe"."personid","e"."companyid",1,"v"."vacancyid",0) IS
NULL
    ORDER BY "currentleave" DESC,
    "pe"."keyName" ASC
END IF
END /* DOC
2019-01-18 PC IW-794 (IW-788) Client - Current Requirements - Shifts ability
to book previous temp
2019-06-19 MHS IW-883
*/
/* TEST
call NetTestSetup('');
def test select * from
NetClientRequirementVerifiedCandidates('TEST.CLIENT','TEST1','1');
```



```
update employment set startdate=current date - 2, leavedate=current date - 1
where personid='TEST';
update person set status='C' where personid='TEST';
insert into tempshiftplan
(tempshiftplanid,vacancyid,description,shiftdate,timefrom,timeto) on
existing update defaults off values ('TEST','TEST','t1',current
date,'09:00','17:00');
insert into tempshift
(tempshiftid,vacancyid,shiftdate,timefrom,timeto,state,tempshiftplanid) on
existing update defaults off values ('TEST','TEST',current
date,'09:00','17:00','B','TEST');
insert into tempshiftplan
(tempshiftplanid,vacancyid,description,shiftdate,timefrom,timeto) on
existing update defaults off values ('TEST1','TEST','t2',current
date,'09:00','17:00');
do test;
expect EOF;
update tempshiftplan set shiftdate=current date + 1 where
tempshiftplanid='TEST1';
do test;
expect PersonID=TEST;
select * from
NetClientRequirementVerifiedCandidates('TEST.CLIENT','TEST','0');
// Just testing execution of non-shift - not result;
*/
GO
```

```
COMMENT TO PRESERVE FORMAT ON PROCEDURE
"pears"."NetClientRequirementVerifiedCandidates" IS
{CREATE PROCEDURE pears."NetClientRequirementVerifiedCandidates"(IN
pWebUserID CHAR(20),IN pRowID CHAR(20),IN IsShift CHAR(1))
RESULT(PersonName CHAR(60),PersonID CHAR(20),PersonStatus CHAR(20),StartDate
CHAR(20),LeaveDate CHAR(20),CurrentLeave DATE)
// IQXWeb
BEGIN
DECLARE currentStates CHAR(100);
DECLARE LOCAL TEMPORARY TABLE Cos(
    CompanyID CHAR(20) NULL,
    ) NOT transactional;
INSERT INTO Cos( CompanyID )
    SELECT companyid FROM employment KEY JOIN ixqnetuserlink WHERE
ixqnetuserid = pWebUserID;
SET currentStates=(SELECT FIRST isnull(personcurrentstates,'C') FROM params
ORDER BY dbid);
IF IsShift='1' THEN
    SELECT
        pe.name,
        pe.personid,
```



```
    st.name AS PersonStatus,
    (SELECT MIN(shiftdate) FROM tempshift WHERE vacancyid=v.vacancyid AND
personid=pe.personid AND state<>'C') AS StartDate,
    (SELECT MAX(shiftdate) FROM tempshift WHERE vacancyid=v.vacancyid AND
personid=pe.personid AND state<>'C') AS LeaveDate,
    LeaveDate AS currentleave
FROM
    tempshiftplan s
    KEY JOIN vacancy v
    KEY JOIN employment e,
    person pe
    JOIN STATUS st ON st.status=pe.status AND st.type='P'
WHERE
    e.companyid IN (SELECT companyid FROM cos)
    AND s.tempshiftplanid=pRowID
    AND pe.personid IN (SELECT personid FROM tempshift WHERE
vacancyid=v.vacancyid AND state<>'C')
    AND locate(currentStates,pe.status)>0
    AND EmployeeAcceptable(pe.personid,e.companyid,1,v.vacancyid,0) IS
NULL
    AND
isnull(personavailableforplan(pe.personid,s.ShiftDate,s.TimeFrom,s.TimeTo,s.
RecoveryHours,s.minutes,0,v.departmentid,s.essentialskill,s.essentialskillgr
adeid,s.essentialskillchoicelist,NULL),'A') LIKE '[AM]%'
ORDER BY
    currentleave DESC,
    pe.KeyName ASC
ELSE
SELECT
    pe.name,
    pe.personid,
    st.name AS PersonStatus,
    (SELECT MAX(startdate) FROM employment WHERE personid=pe.personid) AS
StartDate,
    (SELECT MAX(leavedate) FROM employment WHERE personid=pe.personid) AS
"LeaveDate",
    isnull(e.leavedate, CURRENT DATE) AS currentleave
FROM
    vacancy v
    KEY JOIN employment e,
    person pe
    JOIN STATUS st ON st.status=pe.status AND st.type='P'
WHERE
    e.companyid IN (SELECT companyid FROM cos)
    AND v.vacancyid=pRowID
    AND pe.personid IN (SELECT emp.personid FROM employment emp KEY JOIN
placement pl WHERE pl.vacancyid=v.vacancyid)
    AND locate(currentStates,pe.status)>0
```



```
        AND EmployeeAcceptable(pe.personid,e.companyid,1,v.vacancyid,0) IS
NULL
    ORDER BY
        currentleave DESC,
        pe.keyName ASC
END IF;
END

/* DOC
2019-01-18 PC IW-794 (IW-788) Client - Current Requirements - Shifts ability
to book previous temp
2019-06-19 MHS IW-883
*/

/* TEST
call NetTestSetup('');
def test select * from
NetClientRequirementVerifiedCandidates('TEST.CLIENT','TEST1','1');
update employment set startdate=current date - 2, leavedate=current date - 1
where personid='TEST';
update person set status='C' where personid='TEST';
insert into tempshiftplan
(tempshiftplanid,vacancyid,description,shiftdate,timefrom,timeto) on
existing update defaults off values ('TEST','TEST','t1',current
date,'09:00','17:00');
insert into tempshift
(tempshiftid,vacancyid,shiftdate,timefrom,timeto,state,tempshiftplanid) on
existing update defaults off values ('TEST','TEST',current
date,'09:00','17:00','B','TEST');
insert into tempshiftplan
(tempshiftplanid,vacancyid,description,shiftdate,timefrom,timeto) on
existing update defaults off values ('TEST1','TEST','t2',current
date,'09:00','17:00');
do test;
expect EOF;
update tempshiftplan set shiftdate=current date + 1 where
tempshiftplanid='TEST1';
do test;
expect PersonID=TEST;
select * from
NetClientRequirementVerifiedCandidates('TEST.CLIENT','TEST','0');
// Just testing execution of non-shift - not result;
*/
}
```



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

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

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

