



pears.NetClientPermanentRequirements



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

Original SQL

```
CREATE PROCEDURE "pears"."NetClientPermanentRequirements"(  
  IN "pWebUserID" CHAR(20),  
  IN "pSlice" INTEGER DEFAULT 0,  
  IN "pSliceSize" INTEGER DEFAULT 10,  
  IN "pIncCancelled" SMALLINT DEFAULT 0,  
  IN "pStartDate" CHAR(26) DEFAULT NULL,  
  IN "pEndDate" CHAR(26) DEFAULT NULL,  
  IN "SearchString" CHAR(100) DEFAULT NULL,  
  IN "pStatus" CHAR(1) DEFAULT NULL )  
RESULT(  
  "VacancyID" CHAR(20),  
  "StartDate" CHAR(20),  
  "Position" CHAR(50),  
  "CompanyID" CHAR(20),  
  "CompanyName" CHAR(60),  
  "ContactName" CHAR(60),  
  "Consultant" CHAR(60),  
  "TheirRef" CHAR(50),  
  "ContractRef" CHAR(20),  
  "RefCode" CHAR(20),  
  "FilledBy" CHAR(250),  
  "FinishDate" CHAR(30),  
  "LongDescription" long VARCHAR,  
  "TextStatus" CHAR(100),  
  "CreatedBy" CHAR(255),  
  "VacancyNotes" long VARCHAR,  
  "TextStatusFull" CHAR(100),  
  "SiteDetails" CHAR(300),  
  "NoOfPlacements" INTEGER,  
  "NoOfActivePlacements" INTEGER )  
BEGIN  
  DECLARE "DateRangeStart" DATE =  
  "isnull"("IQXNetStringToDate"("pStartDate"), "dateadd"("week", -1, CURRENT  
DATE));  
  DECLARE "DateRangeEnd" DATE =
```



```
"isnull"("IQXNetStringToDate"("pEndDate"),"dateadd"("year",1,CURRENT DATE));
DECLARE "SendVNotesToWeb" CHAR(10) =
"isnull"("WPKMaintainGetSwitchValue"('SENDVNOTESTOWEB','WEBSITE','l'),'');
DECLARE "SendVOtherNotesToWeb" CHAR(10) =
"isnull"("WPKMaintainGetSwitchValue"('SENDVOTHERNOTESTOWEB','WEBSITE','l'),'');
DECLARE LOCAL TEMPORARY TABLE "IDs"(
  "ID" CHAR(20) NOT NULL,
  "CompanyStartDate" DATE NULL,
) NOT transactional;
INSERT INTO "IDs"("ID","CompanyStartDate")
  SELECT DISTINCT "companyid","IQXNetCompanyStartDate"("companyid") FROM
"employment" KEY JOIN "iqxnetuserlink" WHERE "iqxnetuserid" = "pWebUserID";
IF "IQXNetHasPermission"("pWebUserID",'ORDERJOBS') = 0 THEN
  RETURN
END IF;
SELECT top "pSliceSize" START at("pSliceSize"*"pSlice"+1)
  "v"."VacancyID",
  "v"."Startdate",
  "v"."Position",
  "Company"."CompanyID",
  "Company"."Name" AS "CompanyName",
  "Person"."Name" AS "ContactName",
  "st"."Name" AS "Consultant",
  "v"."TheirRef",
  "v"."ContractRef",
  "v"."Refcode",
  IF("isnull"("v"."noofposts",0) <= 1) THEN
    "isnull"((SELECT FIRST "ep"."name" FROM "placement" KEY JOIN
"employment" AS "em" KEY JOIN "person" AS "ep" WHERE "placement"."vacancyid"
= "v"."vacancyid" AND CURRENT DATE BETWEEN "em"."startdate" AND
"isnull"("em"."leavedate",CURRENT DATE)), 'Unfilled')
  ELSE
    "string"("v"."noofposts",' Posts, ',
    "NoOfPlacements",' Placements, ',
    "NoOfActivePlacements",' Active')
  endif AS "FilledBy",
  IF("v"."noofposts" = 1) THEN
    "isnull"((SELECT FIRST "em"."leavedate" FROM "placement" KEY JOIN
"employment" AS "em" KEY JOIN "person" AS "ep" WHERE "placement"."vacancyid"
= "v"."vacancyid" AND CURRENT DATE BETWEEN "em"."startdate" AND
"isnull"("em"."leavedate",CURRENT DATE) AND "em"."leavedate" IS NOT
NULL),"v"."finishdate")
  ELSE
    "v"."finishdate"
  endif AS "FinishDate",
  (IF "SendVOtherNotesToWeb" = 'Y' THEN "v"."OtherNotes" ELSE '' endif) AS
"LongDescription",
```



```
(IF "NoOfActivePlacements" = 0 THEN 'Unfilled' ELSE IF
"NoOfActivePlacements" < "v"."noofposts" THEN 'Partially filled' ELSE
"s"."name" endif endif) AS "TextStatus",
    "string"('Created by: ', "st"."name", ' at:
', "dateformat"("v"."whenentered", 'dd/mm/yyyy hh:mm')) AS "CreatedBy",
    (IF "SendVNotesToWeb" = 'Y' THEN "v"."notes" ELSE '' endif) AS
"VacancyNotes",
    "TextStatus" AS "TextStatusFull",
    "isnull"("string"("v"."sitename", (IF "isnull"("v"."addr1", '') <> ''
THEN (IF "v"."Sitename" <> '' THEN ', ' endif) + "v"."addr1" endif), (IF
"isnull"("v"."addr2", '') <> '' THEN ', ' + "v"."addr2" endif), (IF
"isnull"("v"."addr3", '') <> '' THEN ', ' + "v"."addr3" endif), (IF
"isnull"("v"."Town", '') <> '' THEN ', ' + "v"."town" endif), (IF
"isnull"("v"."postcode", '') <> '' THEN ', ' + "v"."postcode"
endif))), "GetCompanyAddressOnLine"("e"."companyid")) AS "SiteDetails",
    "isnull"((SELECT "count"() FROM "placement" KEY JOIN "employment" AS
"em" KEY JOIN "person" AS "ep" WHERE "placement"."vacancyid" =
"v"."vacancyid" AND "isnull"("placement"."withdrawreason", '') = ''), 0) AS
"NoOfPlacements",
    (SELECT "count"() FROM "placement" KEY JOIN "employment" AS "em" KEY
JOIN "person" AS "ep" WHERE "placement"."vacancyid" = "v"."vacancyid" AND
CURRENT DATE BETWEEN "isnull"("em"."startdate", CURRENT DATE) AND
"isnull"("em"."leavedate", CURRENT DATE) AND
"isnull"("placement"."withdrawreason", '') = '') AS "NoOfActivePlacements"
FROM "vacancy" AS "v" KEY JOIN "Employment" AS "e" KEY
JOIN("Company", "Person")
    JOIN "IDs" ON "e"."CompanyID" = "IDs"."ID"
    , "vacancy" AS "v" KEY LEFT OUTER JOIN "Staff" AS "st"
    , "vacancy" AS "v" LEFT OUTER JOIN "Status" AS "s" ON "s"."Status" =
"v"."Status" AND "s"."Type" = 'V'
    , "vacancy" AS "v" KEY JOIN("tempdesk", "Department" AS "d")
WHERE "StartDate" <= "DateRangeEnd"
AND "v"."status" <> '['
AND "tempdesk"."desktype" <> 'S'
AND "v"."status" = "isnull"("pStatus", "v"."status")
AND "isnull"("v"."expiry", "today"()) >= "today"()
AND "isnull"("vacancy"."temp", 0) = 0
AND ("pIncCancelled" = 0 AND "v"."status" <> 'X'
OR "pIncCancelled" = 1)
ORDER BY "StartDate" DESC, "v"."VacancyID" DESC
END /* DOC
2020-07-06 ET Created based on NetClientRequirmentList modified to only
include permanent requirements
*/
GO

COMMENT TO PRESERVE FORMAT ON PROCEDURE
"pears"."NetClientPermanentRequirements" IS
```



```
{CREATE PROCEDURE pears."NetClientPermanentRequirements"(
  IN pWebUserID CHAR(20),
  IN pSlice INTEGER DEFAULT 0,
  IN pSliceSize INTEGER DEFAULT 10,
  IN pIncCancelled SMALLINT DEFAULT 0,
  IN pStartDate CHAR(26) DEFAULT NULL,
  IN pEndDate CHAR(26) DEFAULT NULL,
  IN SearchString CHAR(100) DEFAULT NULL,
  IN pStatus CHAR(1) DEFAULT NULL
)
RESULT(
  VacancyID CHAR(20),
  StartDate CHAR(20),
  POSITION CHAR(50),
  CompanyID CHAR(20),
  CompanyName CHAR(60),
  ContactName CHAR(60),
  Consultant CHAR(60),
  TheirRef CHAR(50),
  ContractRef CHAR(20),
  RefCode CHAR(20),
  FilledBy CHAR(250),
  FinishDate CHAR(30),
  LongDescription long VARCHAR,
  TextStatus CHAR(100),
  CreatedBy CHAR(255),
  VacancyNotes long VARCHAR,
  TextStatusFull CHAR(100),
  SiteDetails CHAR(300),
  NoOfPlacements INTEGER,
  NoOfActivePlacements INTEGER
)
BEGIN
  DECLARE DateRangeStart DATE =
isnull(IQXNetStringToDate(pStartDate),dateadd(week,-1,CURRENT DATE));
  DECLARE DateRangeEnd DATE =
isnull(IQXNetStringToDate(pEndDate),dateadd(YEAR,1,CURRENT DATE));
  DECLARE SendVNotesToWeb CHAR(10) =
isnull(WPKMaintainGetSwitchValue('SENDVNOTESTOWEB','WEBSITE','1'),'');
  DECLARE SendVOtherNotesToWeb CHAR(10) =
isnull(WPKMaintainGetSwitchValue('SENDVOTHERNOTESTOWEB','WEBSITE','1'),'');

  DECLARE LOCAL TEMPORARY TABLE IDs(
    ID CHAR(20) NOT NULL,
    CompanyStartDate DATE NULL,
  ) NOT TRANSACTIONAL;

  INSERT INTO IDs( ID,CompanyStartDate )
```



```
SELECT DISTINCT companyid,IQXNetCompanyStartDate(companyid) FROM
employment KEY JOIN iqxnetuserlink WHERE iqxnetuserid = pWebUserID;

IF IQXNetHasPermission(pWebUserID,'ORDERJOBS') = 0 THEN
RETURN
END IF;

SELECT TOP pSliceSize START AT (pSliceSize*pSlice+1)
v.VacancyID,
v.Startdate,
v.Position,
Company.CompanyID,
Company.Name AS CompanyName,
Person.Name AS ContactName,
st.Name AS Consultant,
v.TheirRef,
v.ContractRef,
v.Refcode,
IF(isnull(v.noofposts,0) <= 1) THEN
    isnull((SELECT FIRST ep.name FROM placement KEY JOIN employment AS em
KEY JOIN person AS ep WHERE placement.vacancyid = v.vacancyid AND CURRENT
DATE BETWEEN em.startdate AND isnull(em.leavedate,CURRENT DATE)), 'Unfilled')
ELSE
    string(v.noofposts, ' Posts, ',
    NoOfPlacements, ' Placements, ',
    NoOfActivePlacements, ' Active')
endif AS FilledBy,
IF(v.noofposts = 1) THEN
    isnull((SELECT FIRST em.leavedate FROM placement KEY JOIN employment
AS em KEY JOIN person AS ep WHERE placement.vacancyid = v.vacancyid AND
CURRENT DATE BETWEEN em.startdate AND isnull(em.leavedate,CURRENT DATE) AND
em.leavedate IS NOT NULL),v.finishdate)
ELSE
    v.finishdate
endif AS FinishDate,
(IF SendV0therNotesToWeb = 'Y' THEN v.OtherNotes ELSE '' endif) AS
LongDescription,
(IF NoOfActivePlacements = 0 THEN 'Unfilled' ELSE IF
NoOfActivePlacements < v.noofposts THEN 'Partially filled' ELSE s.name endif
endif) AS TextStatus,
string('Created by: ',st.name, ' at:
',dateformat(v.whenentered,'dd/mm/yyyy hh:mm')) AS CreatedBy,
(IF SendVNotesToWeb = 'Y' THEN v.notes ELSE '' endif) AS VacancyNotes,
TextStatus AS TextStatusFull,
isnull(string(v.sitename,(IF isnull(v.addr1,'') <> '' THEN(IF v.Sitename
<> '' THEN ', ' endif)+v.addr1 endif),(IF isnull(v.addr2,'') <> '' THEN ',
'+v.addr2 endif),(IF isnull(v.addr3,'') <> '' THEN ', '+v.addr3 endif),(IF
```



```
isnull(v.Town,'') <> '' THEN ', '+v.town endif),(IF isnull(v.postcode,'') <>
'' THEN ', '+v.postcode endif)),GetCompanyAddressOnLine(e.companyid)) AS
SiteDetails,
    isnull((SELECT COUNT() FROM placement KEY JOIN employment AS em KEY JOIN
person AS ep WHERE placement.vacancyid = v.vacancyid AND
isnull(placement.withdrawreason,'') = ''),0) AS NoOfPlacements,
    (SELECT COUNT() FROM placement KEY JOIN employment AS em KEY JOIN person
AS ep WHERE placement.vacancyid = v.vacancyid AND CURRENT DATE BETWEEN
isnull(em.startdate,CURRENT DATE) AND isnull(em.leavedate,CURRENT DATE) AND
isnull(placement.withdrawreason,'') = '') AS NoOfActivePlacements
FROM vacancy AS v KEY JOIN Employment AS e KEY JOIN (Company,Person)
JOIN IDs ON e.CompanyID = IDs.ID
, vacancy AS v KEY LEFT OUTER JOIN Staff AS st
, vacancy AS v LEFT OUTER JOIN STATUS AS s ON s.Status = v.Status AND
s.Type = 'V'
, vacancy AS v KEY JOIN (tempdesk, Department AS d)
WHERE
    StartDate <= DateRangeEnd
    AND v.status <> '['
    AND tempdesk.desktype <> 'S'
    AND v.status = isnull(pStatus,v.status)
    AND isnull(v.expiry,today()) >= today()
    AND isnull(vacancy.temp,0) = 0
    AND (
        pIncCancelled = 0 AND v.status <> 'X'
        OR
        pIncCancelled = 1
    )
ORDER BY StartDate DESC, v.VacancyID DESC
END

/* DOC
    2020-07-06 ET Created based on NetClientRequirmentList modified to only
include permanent requirements
*/
}
```

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

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

Last update: 2026/08/07 19:24

