



pears.NetClientRequirementList



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

Original SQL

```
CREATE PROCEDURE "pears"."NetClientRequirementList"( 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( "RowID" CHAR(20),"StartDate" CHAR(20),"CombinedTime"
CHAR(20),"IsShift" SMALLINT,"Position" CHAR(50),"ShiftDescription"
CHAR(50),"CompanyID" CHAR(20),
"CompanyName" CHAR(60),"ContactName" CHAR(60),"Consultant"
CHAR(60),"OurRef" CHAR(50),"ContractRef" CHAR(20),"ShiftRef" CHAR(20),
"FilledBy" CHAR(250),"StartTime" CHAR(15),"FinishTime" CHAR(15),
"FinishDate" CHAR(30),"LongDescription" long VARCHAR,"PersonID" CHAR(20),
"TextStatus" CHAR(100),"CreatedBy" CHAR(255),"CancelState"
CHAR(50),"ReqDescription" CHAR(10),"ClientNote" CHAR(50),
"VacancyNotes" long VARCHAR,"TextStatusFull" CHAR(100),"SiteDetails"
CHAR(300),"NoOfPlacements" INTEGER,"NoOfActivePlacements" INTEGER )
BEGIN
// IQXNet
DECLARE "DateRangeStart" DATE;
DECLARE "DateRangeEnd" DATE;
DECLARE "x" INTEGER;
DECLARE "y" INTEGER;
DECLARE "SendVNotesToWeb" CHAR(10);
DECLARE "SendVOtherNotesToWeb" CHAR(10);
DECLARE LOCAL TEMPORARY TABLE "IDs"(
"ID" CHAR(20) NOT NULL,
"CompanyStartDate" DATE NULL,
) NOT transactional;
IF "IQXNetHasPermission"("pWebUserID",'ORDERJOBS') = 0 THEN
RETURN
END IF;
SELECT
"isnull"("WPKMaintainGetSwitchValue"('SENDVNOTESTOWEB','WEBSITE','l'),'')
INTO "SendVNotesToWeb";
SELECT
```



```
"isnull"("WPKMaintainGetSwitchValue"('SENDVOTHERNOTESTOWEB','WEBSITE','l'),'')
') INTO "SendVOtherNotesToWeb";
INSERT INTO "IDs"("ID","CompanyStartDate")
SELECT DISTINCT "companyid","IQXNetCompanyStartDate"("companyid") FROM
"employment" KEY JOIN "iqxnetuserlink" WHERE "iqxnetuserid" = "pWebUserID";
SET "DateRangeStart" = "IQXNetStringToDate"("pStartDate");
SET "DateRangeEnd" = "IQXNetStringToDate"("pEndDate");
SET "DateRangeStart" =
"isnull"("DateRangeStart","dateadd"("week",-1,CURRENT DATE));
SET "DateRangeEnd" = "isnull"("DateRangeEnd","dateadd"("year",1,CURRENT
DATE));
SET "x" = "pSliceSize";
SET "y" = "x"*"pSlice"+1;
SELECT top "x" START at "y" *
FROM(SELECT "v"."VacancyID" AS "RowID",
"v"."startdate" AS "StartDate",
NULL AS "CombinedTime",
0 AS "IsShift",
"v"."Position" AS "Position",
NULL AS "ShiftDescription",
"Company"."companyID" AS "CompanyID",
"Company"."Name" AS "CompanyName",
"Person"."Name" AS "ContactName",
"st"."Name" AS "Consultant",
"v"."TheirRef" AS "OurRef",
"v"."ContractRef",
"v"."refcode" AS "ShiftRef",
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",
NULL AS "StartTime",
NULL AS "FinishTime",
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",
    '' AS "PersonID",
    (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",
    NULL AS "CancelState",
    'Contract' AS "ReqDescription",
    '' AS "ClientNote",
    (IF "SendVNNotesToWeb" = '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 ("pIncCancelled" = 0 AND "v"."status" <> 'X'
OR "pIncCancelled" = 1) UNION ALL
SELECT "p"."TempShiftPlanID" AS "RowID",
    "isnull"("s"."shiftdate", "p"."ShiftDate", "v"."startdate") AS
"StartDate",
    CAST("dateformat"("StartTime", 'hh:nn') AS CHAR) + ' -
```



```
' + CAST("dateformat"("FinishTime", 'hh:nn') AS CHAR) AS "CombinedTime",
1 AS "IsShift",
"v"."Position" AS "Position",
"p"."Description" AS "ShiftDescription",
"Company"."companyID" AS "CompanyID",
"Company"."Name" AS "CompanyName",
"Pe"."Name" AS "ContactName",
"staff"."name" AS "Consultant",
"isnull"("s"."referencecode", "p"."referencecode") AS "OurRef",
"v"."ContractRef",
CAST("p"."ShiftSerialNumber" AS CHAR(20)) AS "ShiftRef",
"string"("pt"."name", (IF "pt"."status" = 'A' THEN ' (S)' endif)) AS
"FilledBy",
"dateformat"("isnull"("s"."timefrom", "p"."TimeFrom"), 'hh:mm') AS
"StartTime",
"dateformat"("isnull"("s"."timeto", "p"."TimeTo"), 'hh:mm') AS
"FinishTime",
(IF "FinishTime" < "StartTime" THEN "dateadd"("dd", 1, "StartDate")
ELSE "StartDate" endif) AS "FinishDate",
(IF "SendVOtherNotesToWeb" <> 'Y' THEN ''
ELSE IF "isnull"("d"."showvacextranotes", 0) = 1 THEN
CAST("v"."extranotes" AS long VARCHAR)
ELSE CAST("v"."othernotes" AS long VARCHAR)
endif
endif) AS "LongDescription", "s"."personid" AS "PersonID",
"NetTempShiftStatusString"("s"."state") AS "TextStatus",
('Ordered by: ' + "p"."orderby" + ' at:
' + "dateformat"("p"."whenentered", 'dd/mm/yyyy hh:mm')) AS "CreatedBy",
"string"("dateformat"("s"."WhenCancelled", 'dd/mm/yyyy hh:nn:ss'), '
by ', "isnull"((SELECT "name" FROM "staff" WHERE "staff"."staffid" =
"s"."WhoCancelled"), '(not recorded)')) AS "CancelState",
'Shift' AS "ReqDescription",
"p"."ClientNote",
(IF "SendVNotesToWeb" = 'Y' THEN "v"."notes" ELSE '' endif) AS
"VacancyNotes",
"NetTempShiftStatusString"("s"."state", "s"."tempconfirmed", "s"."clientconfir
med", "cr"."name") 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")),
0 AS "NoOfPlacements",
0 AS "NoOfActivePlacements"
FROM "TempShiftplan" AS "p"
KEY LEFT OUTER JOIN "tempshift" AS "s"
```



```
, "TempShiftplan" AS "p" KEY JOIN "vacancy" AS "v" KEY JOIN
"tempdesk" AS "td"
KEY JOIN "Employment" AS "e"
KEY JOIN("Company", "Person" AS "Pe")
JOIN "IDs" ON "e"."CompanyID" = "IDs"."ID"
, "vacancy" AS "v" KEY JOIN "Department" AS "d"
, "vacancy" AS "v" KEY LEFT OUTER JOIN "Staff"
, "tempshift" AS "s" KEY LEFT OUTER JOIN "person" AS "pt"
, "tempshift" AS "s" KEY LEFT OUTER JOIN "tempshiftcancelreason" AS
"cr"

WHERE "StartDate" BETWEEN "DateRangeStart" AND "DateRangeEnd"
AND "v"."status" = "isnull"("pStatus", "v"."status")
AND "td"."desktype" = 'S'
AND "StartDate" >= "IDs"."CompanyStartDate"
AND ("pIncCancelled" = 0 AND NOT("isnull"("s"."State", '') = 'C' AND
"isnull"("s"."cRefill", 0) = 0)
OR "pIncCancelled" = 1)) AS "innerquery"
ORDER BY "StartDate" DESC, "RowID" DESC
END /* DOC
2016-10-21 PC IW-71 Create list of Current Requirements for Client.
SearchString not implemented.
2018-02-28 PC IW-388 Client reference code
2018-03-12 PC IW-390 changes by JB incorporated
2018-04-13 JB iw-410 return vacancy notes, full status, change cancel
2018-05-30 PC IW-588 modify shift status to include candidate and client
confirmed, and other changes to simplify linking
2018-06-06 MHS reduced and simplified, got TextStatusFull working. Fixed
tests.
2019-01-10 PC IW-622 adjust visibility of permanent requirements
2019-02-19 PC IW-839 modify ReqDescription to check for tempdesk
2019-02-19 PC IW-825 add site details and clients details columns
2019-02-21 PC IW-825 modify address to refer to site address first
2019-02-22 PC IW-825 adjust temp filter to look for only non-temp
2019-02-27 PC IW-825 split temps into contract/shift
2019-05-03 MHS fixed to show Contract temp vacancies which had been
erroneously excluded
2019-05-03 MHS Removed weird hybrid contract/shift query. Fixed erroneous
outer join. Extended tests to cover shifts.
2020-01-31 AW removed seconds from date formatting on ln 105 + ln 63
2020-03-16 GJ/MHS Added include cancelled switch and expanded the tests
2020-05-27 GJ IW-1187 Added companyID to result set
2020-06-09 MHS put the union into an inner query and the slicing and
ordering on the outer query (adding RowID to the order by so that it is more
deterministic), so that slicing works properly
*/
/* TEST
call NetTestSetup('');
delete from tempshift from tempshift key join vacancy where
```



```
vacancy.employmentid in (select employmentid from iqxnetuserlink where
iqxnetuserid='test.client');
delete from tempshiftplan from tempshiftplan key join vacancy where
vacancy.employmentid in (select employmentid from iqxnetuserlink where
iqxnetuserid='test.client');
update vacancy set startdate=current date-100,finishdate=current
date-100,expiry=current date-100 where employmentid='TEST' and
vacancyid<>'TEST';
update vacancy set status='C',startdate=current date,finishdate=current
date+5 where vacancyid='TEST';
update tempdesk set desktype='W' where tempdeskid='test';
select * from NetClientRequirementList(null,0,9);
expect EOF;
delete from iqxnetrightallowed where iqxnetrightid='ORDERJOBS' and
IQXNetUserClassID='CLIENT';
select * from NetClientRequirementList('test.client',0,10);
expect EOF;
insert into iqxnetrightallowed
(iqxnetrightid,iqxnetuserclassid,id,maintainonweb) values
('ORDERJOBS','CLIENT','*',1);
select * from NetClientRequirementList('test.client',0,11);
expect CompanyName=test, isshift=0;
select * from
NetClientRequirementList('test.client',0,12,1,dateformat(current
date+10,'dd/mm/yyyy'),dateformat(current date+20,'dd/mm/yyyy'));
expect CompanyName=test, isshift=0;
select * from NetClientRequirementList('test.client',0,13,1,current
date-10,current date+20);
expect CompanyName=test;
update vacancy set status='X' where vacancyid='TEST';
select * from NetClientRequirementList('test.client',0,131,1,current
date-10,current date+20);
expect CompanyName=test;
select * from NetClientRequirementList('test.client',0,132,0,current
date-10,current date+20);
expect EOF;
update tempdesk set desktype='S' where tempdeskid='test';
select * from NetClientRequirementList('test.client',0,14) where isshift=1;
expect EOF;
insert into tempshiftplan(tempshiftplanid,vacancyid,description,shiftdate)
values ('TEST','TEST','TestShift',current date);
select * from NetClientRequirementList('test.client',0,15) where isshift=1;
expect ShiftDescription=TestShift;
insert into
tempshift(tempshiftid,tempshiftplanid,vacancyid,shiftdate,state,personid)
values ('TEST','TEST','TEST',current date,'P','TEST');
select * from NetClientRequirementList('test.client',0,16) where isshift=1;
expect ShiftDescription=TestShift;
```



```
update tempshift set state='C' where tempshiftid='TEST';
select * from NetClientRequirementList('test.client',0,17) where isshift=1;
expect EOF;
select * from NetClientRequirementList('test.client',0,18,1) where
isshift=1;
expect ShiftDescription=TestShift;
update tempshift set crefill=1 where tempshiftid='TEST';
select * from NetClientRequirementList('test.client',0,19) where isshift=1;
expect ShiftDescription=TestShift;
â€œ<
*/
GO
```

COMMENT TO PRESERVE FORMAT ON PROCEDURE "pears"."NetClientRequirementList"
IS

```
{CREATE PROCEDURE pears."NetClientRequirementList"( 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( RowID CHAR(20),StartDate CHAR(20),CombinedTime CHAR(20),IsShift
SMALLINT,POSITION CHAR(50),ShiftDescription CHAR(50), CompanyID CHAR(20),
    CompanyName CHAR(60),ContactName CHAR(60),Consultant CHAR(60),OurRef
CHAR(50),ContractRef CHAR(20),ShiftRef CHAR(20),
    FilledBy CHAR(250),StartTime CHAR(15),FinishTime CHAR(15),
    FinishDate CHAR(30),LongDescription long VARCHAR,PersonID CHAR(20),
    TextStatus CHAR(100),CreatedBy CHAR(255),CancelState
CHAR(50),ReqDescription CHAR(10),ClientNote CHAR(50),
    VacancyNotes long VARCHAR,TextStatusFull CHAR(100),SiteDetails
CHAR(300),NoOfPlacements INTEGER,NoOfActivePlacements INTEGER )
BEGIN
    // IQXNet
    DECLARE DateRangeStart DATE;
    DECLARE DateRangeEnd DATE;
    DECLARE x INTEGER;
    DECLARE y INTEGER;
    DECLARE SendVNotesToWeb CHAR(10);
    DECLARE SendVOtherNotesToWeb CHAR(10);
    DECLARE LOCAL TEMPORARY TABLE IDs(
        ID CHAR(20) NOT NULL,
        CompanyStartDate DATE NULL,
    ) NOT transactional;
    IF IQXNetHasPermission(pWebUserID,'ORDERJOBS') = 0 THEN
        RETURN
    END IF;
    SELECT
isnull(WPKMaintainGetSwitchValue('SENDVNOTESTOWEB','WEBSITE','1'),'') INTO
SendVNotesToWeb;
```




```
SELECT
isnull(WPKMaintainGetSwitchValue('SENDVOTHERNOTESTOWEB','WEBSITE','1'),'')
INTO SendVOtherNotesToWeb;
INSERT INTO IDs( ID,CompanyStartDate )
    SELECT DISTINCT companyid,IQXNetCompanyStartDate(companyid) FROM
employment KEY JOIN iqxnetuserlink WHERE iqxnetuserid = pWebUserID;
SET DateRangeStart = IQXNetStringToDate(pStartDate);
SET DateRangeEnd = IQXNetStringToDate(pEndDate);
SET DateRangeStart = isnull(DateRangeStart,dateadd(week,-1,CURRENT DATE));
SET DateRangeEnd = isnull(DateRangeEnd,dateadd(YEAR,1,CURRENT DATE));
SET x = pSliceSize;
SET y = x*pSlice+1;
SELECT top x START at y * FROM (SELECT
    v.VacancyID AS RowID,
    v.startdate AS StartDate,
    NULL AS CombinedTime,
    0 AS IsShift,
    v.Position AS POSITION,
    NULL AS ShiftDescription,
    Company.companyID AS CompanyID,
    Company.Name AS CompanyName,
    Person.Name AS ContactName,
    st.Name AS Consultant,
    v.TheirRef AS OurRef,
    v.ContractRef,
    v.refcode AS ShiftRef,
    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,
    NULL AS StartTime,
    NULL AS FinishTime,
    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,
    '' AS PersonID,
```




```
(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,
    NULL AS CancelState,
    'Contract' AS ReqDescription,
    '' AS ClientNote,
    (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 (
    pIncCancelled = 0 AND v.status <> 'X'
OR
    pIncCancelled = 1
)
UNION ALL
SELECT p.TempShiftPlanID AS RowID,
    isnull(s.shiftdate,p.ShiftDate,v.startdate) AS StartDate,
    CAST(dateformat(StartTime,'hh:nn') AS CHAR)+' -
'+CAST(dateformat(FinishTime,'hh:nn') AS CHAR) AS CombinedTime,
    1 AS IsShift,
    v.Position AS POSITION,
    p.Description AS ShiftDescription,
    Company.companyID AS CompanyID,
```



```
Company.Name AS CompanyName,
Pe.Name AS ContactName,
staff.name AS Consultant,
isnull(s.referencecode,p.referencecode) AS OurRef,
v.ContractRef,
CAST(p.ShiftSerialNumber AS CHAR(20)) AS ShiftRef,
string(pt.name,(IF pt.status = 'A' THEN ' (S)' endif)) AS FilledBy,
dateformat(isnull(s.timefrom,p.TimeFrom),'hh:mm') AS StartTime,
dateformat(isnull(s.timeto,p.TimeTo),'hh:mm') AS FinishTime,
(IF FinishTime < StartTime THEN dateadd(dd,1,StartDate) ELSE StartDate
endif) AS FinishDate,
(IF SendV0therNotesToWeb <> 'Y' THEN ''
ELSE IF isnull(d.showvacextranotes,0) = 1 THEN CAST(v.extranotes AS long
VARCHAR)
ELSE CAST(v.othernotes AS long VARCHAR)
endif
endif) AS LongDescription,s.personid AS PersonID,
NetTempShiftStatusString(s.state) AS TextStatus,
('Ordered by: '+p.orderedby+' at: '+dateformat(p.whenentered,'dd/mm/yyyy
hh:mm')) AS CreatedBy,
string(dateformat(s.WhenCancelled,'dd/mm/yyyy hh:nn:ss'),' by
',isnull((SELECT name FROM staff WHERE staff.staffid = s.WhoCancelled),'(not
recorded)')) AS CancelState,
'Shift' AS ReqDescription,
p.ClientNote,
(IF SendVNNotesToWeb = 'Y' THEN v.notes ELSE '' endif) AS VacancyNotes,
NetTempShiftStatusString(s.state,s.tempconfirmed,s.clientconfirmed,cr.name)
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)),
0 AS NoOfPlacements,
0 AS NoOfActivePlacements
FROM TempShiftplan AS p
KEY LEFT OUTER JOIN tempshift AS s,
TempShiftplan AS p KEY JOIN vacancy AS v KEY JOIN tempdesk AS td
KEY JOIN Employment AS e
KEY JOIN (Company,Person AS Pe)
JOIN IDs ON e.CompanyID = IDs.ID
,vacancy AS v KEY JOIN Department AS d
,vacancy AS v KEY LEFT OUTER JOIN Staff
,tempshift AS s KEY LEFT OUTER JOIN person AS pt
,tempshift AS s KEY LEFT OUTER JOIN tempshiftcancelreason AS cr
WHERE StartDate BETWEEN DateRangeStart AND DateRangeEnd
AND v.status = isnull(pStatus,v.status)
AND td.desktype = 'S'
```



```
AND StartDate >= IDs.CompanyStartDate
AND (
    pIncCancelled = 0 AND NOT (isnull(s.State,'') = 'C' AND
isnull(s.cRefill,0) = 0)
    OR
    pIncCancelled = 1
) ) AS innerquery
ORDER BY StartDate DESC, RowID DESC
END /* DOC
2016-10-21 PC IW-71 Create list of Current Requirements for Client.
SearchString not implemented.
2018-02-28 PC IW-388 Client reference code
2018-03-12 PC IW-390 changes by JB incorporated
2018-04-13 JB iw-410 return vacancy notes, full status, change cancel
2018-05-30 PC IW-588 modify shift status to include candidate and client
confirmed, and other changes to simplify linking
2018-06-06 MHS reduced and simplified, got TextStatusFull working. Fixed
tests.
2019-01-10 PC IW-622 adjust visibility of permanent requirements
2019-02-19 PC IW-839 modify ReqDescription to check for tempdesk
2019-02-19 PC IW-825 add site details and clients details columns
2019-02-21 PC IW-825 modify address to refer to site address first
2019-02-22 PC IW-825 adjust temp filter to look for only non-temp
2019-02-27 PC IW-825 split temps into contract/shift
2019-05-03 MHS fixed to show Contract temp vacancies which had been
erroneously excluded
2019-05-03 MHS Removed weird hybrid contract/shift query. Fixed erroneous
outer join. Extended tests to cover shifts.
2020-01-31 AW removed seconds from date formatting on ln 105 + ln 63
2020-03-16 GJ/MHS Added include cancelled switch and expanded the tests
2020-05-27 GJ IW-1187 Added companyID to result set
2020-06-09 MHS put the union into an inner query and the slicing and
ordering on the outer query (adding RowID to the order by so that it is more
deterministic), so that slicing works properly
*/
/* TEST
call NetTestSetup('');
delete from tempshift from tempshift key join vacancy where
vacancy.employmentid in (select employmentid from iqxnetuserlink where
iqxnetuserid='test.client');
delete from tempshiftplan from tempshiftplan key join vacancy where
vacancy.employmentid in (select employmentid from iqxnetuserlink where
iqxnetuserid='test.client');
update vacancy set startdate=current date-100,finishdate=current
date-100,expiry=current date-100 where employmentid='TEST' and
vacancyid<>'TEST';
update vacancy set status='C',startdate=current date,finishdate=current
date+5 where vacancyid='TEST';
```



```
update tempdesk set desktype='W' where tempdeskid='test';
select * from NetClientRequirementList(null,0,9);
expect EOF;
delete from iqxnetrightallowed where iqxnetrightid='ORDERJOBS' and
IQXNetUserClassID='CLIENT';
select * from NetClientRequirementList('test.client',0,10);
expect EOF;
insert into iqxnetrightallowed
(iqxnetrightid,ixxnetuserclassid,id,maintainonweb) values
('ORDERJOBS','CLIENT','*',1);
select * from NetClientRequirementList('test.client',0,11);
expect CompanyName=test, isshift=0;
select * from
NetClientRequirementList('test.client',0,12,1,dateformat(current
date+10,'dd/mm/yyyy'),dateformat(current date+20,'dd/mm/yyyy'));
expect CompanyName=test, isshift=0;
select * from NetClientRequirementList('test.client',0,13,1,current
date-10,current date+20);
expect CompanyName=test;
update vacancy set status='X' where vacancyid='TEST';
select * from NetClientRequirementList('test.client',0,131,1,current
date-10,current date+20);
expect CompanyName=test;
select * from NetClientRequirementList('test.client',0,132,0,current
date-10,current date+20);
expect EOF;
update tempdesk set desktype='S' where tempdeskid='test';
select * from NetClientRequirementList('test.client',0,14) where isshift=1;
expect EOF;
insert into tempshiftplan(tempshiftplanid,vacancyid,description,shiftdate)
values ('TEST','TEST','TestShift',current date);
select * from NetClientRequirementList('test.client',0,15) where isshift=1;
expect ShiftDescription=TestShift;
insert into
tempshift(tempshiftid,tempshiftplanid,vacancyid,shiftdate,state,personid)
values ('TEST','TEST','TEST',current date,'P','TEST');
select * from NetClientRequirementList('test.client',0,16) where isshift=1;
expect ShiftDescription=TestShift;
update tempshift set state='C' where tempshiftid='TEST';
select * from NetClientRequirementList('test.client',0,17) where isshift=1;
expect EOF;
select * from NetClientRequirementList('test.client',0,18,1) where
isshift=1;
expect ShiftDescription=TestShift;
update tempshift set crefill=1 where tempshiftid='TEST';
select * from NetClientRequirementList('test.client',0,19) where isshift=1;
expect ShiftDescription=TestShift;
â€œ<
```



```
* /  
}
```

From:

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

Permanent link:

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

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

