



pears.NetOwnerRequirementShifts



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

Original SQL

```
CREATE PROCEDURE "pears"."NetOwnerRequirementShifts"( IN "pWebUserID"
CHAR(20),IN "pSlice" INTEGER DEFAULT 1,IN "pSliceSize" INTEGER DEFAULT 10,IN
"pStartDate" CHAR(26) DEFAULT NULL,IN "pEndDate" CHAR(26) DEFAULT NULL,IN
"pStatus" CHAR(1) DEFAULT NULL,IN "pTempdeskID" CHAR(20) 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),"ContactID"
CHAR(20),"ContactName" CHAR(60),"DetailStatus" CHAR(30),"ListStatus"
CHAR(30),"Department" CHAR(50),"Consultant" CHAR(60),"OurRef"
CHAR(20),"ContractRef" CHAR(20),"ShiftRef" CHAR(20),"WhenEntered"
TIMESTAMP,"FilledBy" CHAR(250),"StartTime" CHAR(15),"FinishTime"
CHAR(15),"FinishDate" CHAR(30),"LongDescription" long
VARCHAR,"CompanyAddress" CHAR(250),"PersonID" CHAR(20),"TextStatus"
CHAR(100),"PONumber" CHAR(50),"CreatedBy" CHAR(255),"ConfirmStatus"
CHAR(250) )
BEGIN
  // IQXWeb
  DECLARE "DateRangeStart" DATE;
  DECLARE "DateRangeEnd" DATE;
  DECLARE "x" INTEGER;
  DECLARE "y" INTEGER;
  DECLARE "DefaultTempdeskID" CHAR(20);
  SET "DateRangeStart" = "IQXNetStringToDate"("pStartDate");
  SET "DateRangeEnd" = "IQXNetStringToDate"("pEndDate");
  SET "DateRangeStart" =
"isnull"("DateRangeStart","dateadd"("week",-1,CURRENT DATE));
  SET "DateRangeEnd" = "isnull"("DateRangeEnd","dateadd"("week",1,CURRENT
DATE));
  SET "x" = "pSliceSize";
  SET "y" = "x"*"pSlice"+1;
  IF NOT EXISTS(SELECT * FROM "staff" KEY JOIN "iqxnetuser" WHERE
"iqxnetuserid" = "pWebUserID" AND "IQXNetUserClassID" = 'OWNER') THEN
    RETURN
  END IF;
  SET "DefaultttempdeskID" = "isnull"("pTempdeskID",(SELECT top 1
```



```
"tempdeskid" FROM "staff" KEY JOIN "iqxnetuser" WHERE "iqxnetuserid" =
"pWebUserID" AND "IQXNetUserClassID" = 'OWNER' ORDER BY "tempdeskid" ASC));
SELECT top "x" START at "y"
    "p"."TempShiftPlanID" AS "RowID",
    "p"."shiftdate" 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",
    "Company"."Name" AS "CompanyName",
    "Pe"."PersonID" AS "ContactID",
    "Pe"."Name" AS "ContactName",
    (SELECT(IF "Status"."final" = 1 OR "ConfirmStatus" IS NULL THEN
"Status"."Name" ELSE "ConfirmStatus" endif) FROM "Status" WHERE
"Status"."Status" = "v"."Status" AND "Status"."Type" = 'V') AS
"DetailStatus",
    (SELECT "Name" FROM "Status" WHERE "Status"."Status" = "v"."Status" AND
"Status"."Type" = 'V') AS "ListStatus",
    (SELECT "Name" FROM "Department" WHERE "Department"."DepartmentID" =
"v"."DepartmentID") AS "Department",
    (SELECT "Name" FROM "Staff" WHERE "Staff"."StaffID" = "v"."StaffID") AS
"Consultant",
    "v"."RefCode" AS "OurRef",
    "v"."ContractRef",
    CAST("p"."ShiftSerialNumber" AS CHAR(20)) AS "ShiftRef",
    "p"."WhenEntered",
    (SELECT "name" FROM "person" WHERE "personid" =
"tes"."personid")+(IF(SELECT "status" FROM "person" WHERE "personid" =
"tes"."personid") = 'A' THEN ' (S)' endif) AS "FilledBy",
    "dateformat"("isnull"("tes"."timefrom","p"."TimeFrom"),'hh:mm') AS
"StartTime",
    "dateformat"("isnull"("tes"."timeto","p"."TimeTo"),'hh:mm') AS
"FinishTime",
    (IF "FinishTime" < "StartTime" THEN "dateadd"("dd",1,"StartDate") ELSE
"StartDate" endif) AS "FinishDate",
    CAST(NULL AS long VARCHAR) AS "LongDescription",
    (IF 1 > 1 THEN "GetCompanyAddressOnLine"("company"."companyid")
ELSE ''
endif) AS "CompanyAddress",
    "tes"."personid" AS "PersonID",
    (IF(SELECT "count"() FROM "contactevent" WHERE "classcode" = 'SS' AND
"description" = "ShiftRef") = 1 THEN 'In Progress'
ELSE(CASE "tes"."state" WHEN 'B' THEN 'Confirmed' WHEN 'W' THEN 'Worked'
WHEN 'P' THEN 'Pending' WHEN 'C' THEN(SELECT "cr"."name" FROM
"TempShiftCancelReason" AS "cr" WHERE "cr"."TempShiftCancelReasonid" =
"tes"."cancelreason")+ ' '+ "tes"."note"+' at
```



```
'+"dateformat"("tes"."whencancelled",'dd/mm/yyyy hh:nn')
    ELSE 'Pending'
    END)
endif) AS "statusText",
    "p"."ReferenceCode" AS "PONumber",
    ('Ordered by: '+'p"."orderby'+ ' at:
'+"dateformat"("p"."whenentered",'dd/mm/yyyy hh:mm:ss')) AS "CreatedBy",
    IF "isnull"("tes"."clientconfirmed",0) = 1 THEN 'Client confirmed'
    ELSE IF "isnull"("tes"."tempconfirmed",0) = 1 THEN 'Candidate confirmed'
    ELSE NULL
    endif
endif AS "ConfirmStatus"
FROM "TempShiftPlan" AS "p"
    KEY JOIN "Vacancy" AS "v"
    KEY JOIN "Employment" AS "e"
    KEY JOIN("Company","Person" AS "Pe")
    , "TempShiftPlan" AS "p"
    KEY LEFT OUTER JOIN "tempshift" AS "tes"
WHERE "StartDate" BETWEEN "DateRangeStart" AND "DateRangeEnd"
AND "v"."status" = "isnull"("pStatus","v"."status")
AND "v"."tempdeskid" = "DefaulttempdeskID"
AND "isnull"("company"."divisionid",'') = any(SELECT "DivisionID" FROM
"DashboardStaffDivisions"())
ORDER BY "StartDate" DESC
END /* DOC
2018-10-08 PC IW-700 Derived from NetOwnerRequirementList
2018-10-11 PC enforce sort order
2018-11-27 IW-758 prioritise selection by tempdesk
*/
/* TEST
update tempshiftplan set shiftdate=current date where
tempshiftplanid='test';
update tempdesk set desktype='S' where tempdeskid in ('test','test2');
update staff set tempdeskid='test' where staffid='test';
update vacancy set status='C';
insert into deptmaintenance (departmentid,staffid) on existing update
defaults off values ('~~','TEST');
select * from
NetOwnerRequirementShifts('test.owner',0,100,null,null,null,null);
expect RowID=TEST,Position=test;
select * from
NetOwnerRequirementShifts('test.owner',0,100,null,null,null,'x99');
expect EOF;
select * from
NetOwnerRequirementShifts('test.owner',0,100,null,null,null,'Test');
expect RowID=TEST,Position=test;
select * from
NetOwnerRequirementShifts('test.owner',0,100,null,null,null,'Test2');
```



```
expect EOF;
select * from
NetOwnerRequirementShifts('test.owner',0,100,null,null,'C',null);
expect RowID=TEST,Position=test;
select * from
NetOwnerRequirementShifts('test.client',0,100,null,null,'Z',null);
expect EOF;
*/
GO
```

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

```
{CREATE PROCEDURE pears."NetOwnerRequirementShifts"( IN "pWebUserID"
CHAR(20),IN "pSlice" INTEGER DEFAULT 1,IN pSliceSize INTEGER DEFAULT 10,IN
"pStartDate" CHAR(26) DEFAULT NULL,IN "pEndDate" CHAR(26) DEFAULT NULL, IN
"pStatus" CHAR(1) DEFAULT NULL, IN pTempdeskID CHAR(20) 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),ContactID
CHAR(20),ContactName CHAR(60),DetailStatus CHAR(30),ListStatus
CHAR(30),Department CHAR(50),"Consultant" CHAR(60),"OurRef"
CHAR(20),"ContractRef" CHAR(20),"ShiftRef" CHAR(20),"WhenEntered"
TIMESTAMP,"FilledBy" CHAR(250),StartTime CHAR(15),FinishTime
CHAR(15),FinishDate CHAR(30),"LongDescription" long VARCHAR,"CompanyAddress"
CHAR(250),"PersonID" CHAR(20),"TextStatus" CHAR(100),"PONumber"
CHAR(50),CreatedBy CHAR(255),ConfirmStatus CHAR(250))
BEGIN
// IQXWeb
    DECLARE DateRangeStart DATE;
    DECLARE DateRangeEnd DATE;
    DECLARE x INTEGER;
    DECLARE y INTEGER;
    DECLARE DefaultTempdeskID CHAR(20);

    SET DateRangeStart=IQXNetStringToDate(pStartDate);
    SET DateRangeEnd=IQXNetStringToDate(pEndDate);
    SET DateRangeStart = "isnull"(DateRangeStart,"dateadd"("week",-1,CURRENT
DATE));
    SET DateRangeEnd = "isnull"(DateRangeEnd,"dateadd"("Week",1,CURRENT
DATE));
    SET x=pSliceSize;
    SET y=x*pSlice+1;

    IF NOT EXISTS(SELECT * FROM staff KEY JOIN iqxnetuser WHERE
iqxnetuserid=pWebUserID AND IQXNetUserClassID ='OWNER') THEN
        RETURN ;
    END IF;
```



```
SET DefaulttempdeskID=isnull(pTempdeskID,(SELECT top 1 tempdeskid FROM
staff KEY JOIN iqxnetuser WHERE iqxnetuserid=pWebUserID AND
IQXNetUserClassID ='OWNER' ORDER BY tempdeskid ASC));

SELECT top x START at y
    "p"."TempShiftPlanID" AS "RowID",
    "p"."shiftdate" 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",
    "Company"."Name" AS "CompanyName",
    "Pe".PersonID AS ContactID,
    "Pe"."Name" AS "ContactName",
    (SELECT (IF STATUS.final=1 OR ConfirmStatus IS NULL THEN
STATUS."Name" ELSE ConfirmStatus endif) FROM "Status" WHERE
"Status"."Status" = "v"."Status" AND "Status"."Type" = 'V') AS DetailStatus,
    (SELECT "Name" FROM "Status" WHERE "Status"."Status" = "v"."Status"
AND "Status"."Type" = 'V') AS "ListStatus",
    (SELECT "Name" FROM "Department" WHERE "Department"."DepartmentID" =
"v"."DepartmentID") AS "Department",
    (SELECT "Name" FROM "Staff" WHERE "Staff"."StaffID" = "v"."StaffID")
AS "Consultant",
    "v"."RefCode" AS "OurRef",
    "v"."ContractRef",
    CAST("p"."ShiftSerialNumber" AS CHAR(20)) AS "ShiftRef",
    "p"."WhenEntered",
    (SELECT "name" FROM "person" WHERE "personid" =
"tes"."personid")+ (IF(SELECT "status" FROM "person" WHERE "personid" =
"tes"."personid") = 'A' THEN ' (S)' endif) AS "FilledBy",
    dateformat("isnull"("tes"."timefrom","p"."TimeFrom"),'hh:mm') AS
"StartTime",
    dateformat("isnull"("tes"."timeto","p"."TimeTo"),'hh:mm') AS
"FinishTime",
    (IF "FinishTime" < "StartTime" THEN "dateadd"("dd",1,"StartDate")
ELSE "StartDate" endif) AS "FinishDate",
    CAST(NULL AS long VARCHAR) AS "LongDescription",
    (IF 1 > 1 THEN "GetCompanyAddressOnLine"("company"."companyid")
ELSE ''
endif) AS "CompanyAddress",
    "tes"."personid" AS "PersonID",
    (IF(SELECT "count"() FROM "contactevent" WHERE "classcode" = 'SS'
AND "description" = "ShiftRef") = 1 THEN 'In Progress'
ELSE(CASE "tes"."state" WHEN 'B' THEN 'Confirmed' WHEN 'W' THEN
'Worked' WHEN 'P' THEN 'Pending' WHEN 'C' THEN(SELECT "cr"."name" FROM
"TempShiftCancelReason" AS "cr" WHERE "cr"."TempShiftCancelReasonid" =
```



```
"tes"."cancelreason")+ ' '+'tes"."note"+' at
+'dateformat'("tes"."whencancelled",'dd/mm/yyyy hh:nn')
    ELSE 'Pending'
    END)
endif) AS "statusText",
"p"."ReferenceCode" AS "PONumber",
('Ordered by: '+'p"."orderedby"+' at:
+'dateformat'("p"."whenentered",'dd/mm/yyyy hh:mm:ss')) AS "CreatedBy",
IF isnull(tes.clientconfirmed,0)=1 THEN 'Client confirmed' ELSE
    IF isnull(tes.tempconfirmed,0)=1 THEN 'Candidate confirmed' ELSE
        NULL
    endif
endif AS "ConfirmStatus"
FROM
    "TempShiftPlan" AS "p"
    KEY JOIN "Vacancy" AS "v"
    KEY JOIN "Employment" AS "e"
    KEY JOIN ("Company","Person" AS "Pe"),
    "TempShiftPlan" AS "p"
    KEY LEFT OUTER JOIN "tempshift" AS "tes"
WHERE
    "StartDate" BETWEEN DateRangeStart AND DateRangeEnd
    AND v.status = isnull(pStatus,v.status)
    AND v.tempdeskid=DefaulttempdeskID
    AND isnull(company.divisionid,'') IN (SELECT DivisionID FROM
DashboardStaffDivisions() )
ORDER BY
    StartDate DESC;
END

/* DOC
2018-10-08 PC IW-700 Derived from NetOwnerRequirementList
2018-10-11 PC enforce sort order
2018-11-27 IW-758 prioritise selection by tempdesk
*/

/* TEST
update tempshiftplan set shiftdate=current date where
tempshiftplanid='test';
update tempdesk set desktype='S' where tempdeskid in ('test','test2');
update staff set tempdeskid='test' where staffid='test';
update vacancy set status='C';
insert into deptmaintenance (departmentid,staffid) on existing update
defaults off values ('~~','TEST');
select * from
NetOwnerRequirementShifts('test.owner',0,100,null,null,null,null);
expect RowID=TEST,Position=test;
select * from
```



```
NetOwnerRequirementShifts('test.owner',0,100,null,null,null,'x99');
expect EOF;
select * from
NetOwnerRequirementShifts('test.owner',0,100,null,null,null,'Test');
expect RowID=TEST,Position=test;
select * from
NetOwnerRequirementShifts('test.owner',0,100,null,null,null,'Test2');
expect EOF;
select * from
NetOwnerRequirementShifts('test.owner',0,100,null,null,'C',null);
expect RowID=TEST,Position=test;
select * from
NetOwnerRequirementShifts('test.client',0,100,null,null,'Z',null);
expect EOF;
*/
}
```

From:

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

Permanent link:

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

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

