



pears.NetCandidateDiaryAdd



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

Original SQL

```
CREATE PROCEDURE "pears"."NetCandidateDiaryAdd"( IN "pWebUserID" CHAR(20)
DEFAULT NULL,IN "pshifttype" CHAR(10) DEFAULT NULL,IN "pdatefrom" CHAR(50)
DEFAULT NULL,IN "pdateto" CHAR(50) DEFAULT NULL,
    IN "ptimefrom" CHAR(50) DEFAULT NULL,IN "ptimeto" CHAR(50) DEFAULT NULL,IN
"IDCode" CHAR(20) DEFAULT NULL,IN "pState" CHAR(2) DEFAULT NULL,IN
"pPersonID" CHAR(20) DEFAULT NULL )
RESULT( "pResult" CHAR(250) )
// IQXWeb
BEGIN
    DECLARE "dfrom" DATE;
    DECLARE "dto" DATE;
    DECLARE "tfrom" TIME;
    DECLARE "tto" TIME;
    SET "pState" = "trim"("isnull"("left"("pState",1),''));
    IF "pState" = 'C' THEN //Cancel
        UPDATE "tempshift" SET "state" = "pState","whencancelled" = CURRENT
TIMESTAMP,"whocancelled" = "userstaffid","crefill" = 1,"clientconfirmed" =
0,"tempconfirmed" = 1,"cancelreason" = 'T'
        WHERE "tempShiftID" = "IDCode" AND "personid" = any(SELECT "personid"
FROM "iqxnetuserlink" WHERE "iqxnetuserid" = "pWebUserID") AND "vacancyid"
IS NOT NULL;
        CALL
"IQXNetRequestAction"("pWebUserID",'CandidateReject','SHIFT',"IDCode");
        SELECT '0:~Success';
        RETURN
    END IF;
    IF "pState" = 'F' THEN //conFirm
        UPDATE "tempshift" SET "tempconfirmed" = 1 WHERE "tempshiftid" =
"IDCode" AND "personid" = any(SELECT "personid" FROM "iqxnetuserlink" WHERE
"iqxnetuserid" = "pWebUserID") AND "vacancyid" IS NOT NULL;
        CALL
"IQXNetRequestAction"("pWebUserID",'CandidateConfirm','SHIFT',"IDCode");
        SELECT '0:~Success';
        RETURN
    END IF;
```



```
IF "pState" = 'D' THEN //DELETE
    DELETE FROM "tempshift" WHERE "tempshiftid" = "IDCode" AND "personid" =
any(SELECT "personid" FROM "iqxnetuserlink" WHERE "iqxnetuserid" =
"pWebUserID") AND "vacancyid" IS NULL AND "state" IN( 'H','A','U' ) ;
    CALL
"IQXNetRequestAction"("pWebUserID",'CandidateDelete','SHIFT',"IDCode");
IF("isnull"("WPKMaintainGetSwitchValue"('AVAILCHECKCONTEVENT','','L'),'') <>
'') THEN
    CALL "NetCreateAvailabilityContactEvent"("pWebUserID",'Candidate
removed availability on the candidate portal.')
    END IF;
    SELECT '0:~Success';
    RETURN
END IF;
SET "pshifttype" = "trim"("isnull"("pshifttype",''));
IF "substring"("pshifttype",2,1) = 'W' THEN -- Whole days
    SET "ptimefrom" = '';
    SET "ptimeto" = ''
END IF;
SET "pshifttype" = "left"("pshifttype",1);
SET "dfrom" = "iqxnetstringtodate"("pdatefrom");
SET "dto" = "iqxnetstringtodate"("pdateto");
SET "tfrom" = "iqxnetstringtotime"("ptimefrom");
SET "tto" = "iqxnetstringtotime"("ptimeto");
IF "IDCode" <> '' AND "pState" = '' THEN
    UPDATE "tempshift" SET "shiftdate" = "dfrom","timefrom" =
"tfrom","timeto" = "tto" WHERE "tempShiftID" = "IDCode" AND "vacancyid" IS
NULL AND "personid" = any(SELECT "personid" FROM "iqxnetuserlink" WHERE
"iqxnetuserid" = "pWebUserID");
    SELECT '0:~Success';
    RETURN
END IF;
SET "pPersonID" = "NetCandidateValidPersonID"("pWebUserID","pPersonID");
IF "pPersonID" IS NULL THEN
    SELECT '99:~Unique candidate link required';
    RETURN
END IF;
IF "pshifttype" NOT IN( 'H','A','U' ) THEN //Holiday/Available/Unavailable
    SELECT '101:~invalid availability type';
    RETURN
END IF;
IF("trim"("isnull"("pdatefrom",'')) <> '' AND "dfrom" IS NULL)
OR("trim"("isnull"("pdateto",'')) <> '' AND "dto" IS NULL)
OR("trim"("isnull"("ptimefrom",'')) <> '' AND "tfrom" IS NULL)
OR("trim"("isnull"("ptimeto",'')) <> '' AND "tto" IS NULL) THEN
    SELECT '102:~Invalid date or time';
    RETURN
END IF;
```



```
IF "dfrom" IS NULL THEN
    SELECT '103:~Invalid date';
    RETURN
END IF;
IF "dto" < "dfrom" THEN
    SELECT '104:~Finish day appears to be before Start day';
    RETURN
END IF;
SET "dto" = "isnull"("dto","dfrom");
IF "tfrom" IS NULL THEN
    SET "tto" = NULL
END IF;
IF "tto" IS NULL THEN
    SET "tfrom" = NULL
END IF;
while "dfrom" <= "dto" loop
    INSERT INTO "tempshift"(
"tempshiftid","personid","shiftdate","timefrom","timeto","state" ) VALUES(
"uniquekey"("dfrom"+"tfrom"),"pPersonID","dfrom","tfrom","tto","pshifttype"
) ;
    SET "dfrom" = "dfrom"+1
END loop;
SELECT '0:~Success'
END /* DOC
```

2016-12-01 PC Doc and test

2018-01-23 PC modify primary key seed to avoid duplicating on several events on same day but different time

2018-05-17 MHS Fix and simplify tests. Allowed fromtime>totime for overnight availability etc.

2018-07-24 PC include improved error handler

2018-10-10 MHS IW-456 multi-cand fix. For shift updates and deletes it only affects them if they belong to the cand. For inserts it insists on unique or explicit pPersonID.

pdatefrom dd/mm/yyyy : Shift date - use when adding or editing holiday, availability or unavailability.

pdateto dd/mm/yyyy : End of date range - use when adding holiday, availability or unavailability.

ptimefrom and

ptimeto hh:mm : use when editing or adding holiday, availability or unavailability. Blank for whole days.

pshifttype: Use when adding A - availability, H - holiday or U - unavailability.

IDCode: The tempShiftID - use when editing or deleting existing holiday, availability or unavailability, or confirming or cancelling a shift. Blank otherwise.

pState: C for cancel shift. F for confirm shift. D for delete holiday, availability or unavailability. Blank otherwise.



Inputs:

pWebUserID
pshifttype
pdatefrom
pdateto
ptimefrom
ptimeto
IDCode
pState

Outputs:

pResult

```
*/
/* TEST
call NetTestSetup('');
delete from tempshift where personid='TEST';
select * from NetCandidateDiaryAdd(null,null,null,null,null,null,null,null);
expect 99*;
select * from
NetCandidateDiaryAdd('test.candidate',null,null,null,null,null,null,null);
expect 101*;
select * from
NetCandidateDiaryAdd('test.candidate','H',null,null,null,null,null,null);
expect 103*;
select * from
NetCandidateDiaryAdd('test.candidate','',null,null,null,null,null,'F');
expect 0*;
select * from
NetCandidateDiaryAdd('test.candidate','',null,null,null,null,null,'C');
expect 0*;
select * from
NetCandidateDiaryAdd('test.candidate','',null,null,null,null,null,'D');
expect 0*;
select * from
NetCandidateDiaryAdd('test.candidate','H',dateformat(now(*),'dd/mm/yyyy'),dateformat(now(*),'dd/mm/yyyy'),dateformat(now(*),'hh:mm:ss'),dateformat(dateadd(dd,1,now(*)),'hh:mm:ss'),null,null);
expect 0*;
select * from tempshift where personid='TEST';
expect State=H;
expect EOF;
*/
GO
```

COMMENT TO PRESERVE FORMAT ON PROCEDURE "pears"."NetCandidateDiaryAdd" IS
{CREATE PROCEDURE pears."NetCandidateDiaryAdd"(IN pWebUserID CHAR(20)



```
DEFAULT NULL,IN pshifttype CHAR(10) DEFAULT NULL,IN pdatefrom CHAR(50)
DEFAULT NULL,IN pdateto CHAR(50) DEFAULT NULL,IN ptimefrom CHAR(50) DEFAULT
NULL,IN ptimeto CHAR(50) DEFAULT NULL,IN IDCode CHAR(20) DEFAULT NULL,IN
pState CHAR(2) DEFAULT NULL, IN pPersonID CHAR(20) DEFAULT NULL)
RESULT(pResult CHAR(250))
// IQXWeb
BEGIN
    DECLARE dfrom DATE;
    DECLARE dto DATE;
    DECLARE tfrom TIME;
    DECLARE ttto TIME;
    SET pState=TRIM(issnull("left"(pState,1),''));
    IF pState = 'C' THEN //Cancel
        UPDATE tempshift SET state = pState,whencancelled = CURRENT
TIMESTAMP,whocancelled = userstaffid,crefill = 1,clientconfirmed =
0,tempconfirmed = 1,cancelreason = 'T'
        WHERE tempShiftID = IDCode AND "personid" IN (SELECT personid FROM
iqxnetuserlink WHERE iqxnetuserid=pWebUserID) AND vacancyid IS NOT NULL;
        CALL IQXNetRequestAction(pWebUserID,'CandidateReject','SHIFT',IDCode);
        SELECT '0:~Success';
        RETURN
    END IF;
    IF pState = 'F' THEN //conFirm
        UPDATE tempshift SET tempconfirmed = 1 WHERE tempshiftid = IDCode AND
"personid" IN (SELECT personid FROM iqxnetuserlink WHERE
iqxnetuserid=pWebUserID) AND vacancyid IS NOT NULL;
        CALL IQXNetRequestAction(pWebUserID,'CandidateConfirm','SHIFT',IDCode);
        SELECT '0:~Success';
        RETURN
    END IF;
    IF pState = 'D' THEN //DELETE
        DELETE FROM tempshift WHERE tempshiftid = IDCode AND "personid" IN
(SELECT personid FROM iqxnetuserlink WHERE iqxnetuserid=pWebUserID) AND
vacancyid IS NULL AND state IN( 'H','A','U' );
        CALL IQXNetRequestAction(pWebUserID,'CandidateDelete','SHIFT',IDCode);
        IF(issnull(WPKMaintainGetSwitchValue('AVAILCHECKCONTEVENT', '', 'L'), ''))
<> '' THEN
            CALL NetCreateAvailabilityContactEvent(pWebUserID, 'Candidate
removed availability on the candidate portal.');
```



```
SET pshifttype="left"(pshifttype,1);
SET dfrom=iqxnetstringtodate(pdatefrom);
SET dto=iqxnetstringtodate(pdateto);
SET tfrom=iqxnetstringtotime(ptimefrom);
SET tto=iqxnetstringtotime(ptimeto);
IF IDCode <> '' AND pState = '' THEN
    UPDATE tempshift SET shiftdate = dfrom,timefrom = tfrom,timeto = tto
WHERE tempShiftID = IDCode AND vacancyid IS NULL AND "personid" IN (SELECT
personid FROM ixqnetuserlink WHERE ixqnetuserid=pWebUserID);
    SELECT '0:~Success';
    RETURN
END IF;
SET pPersonID=NetCandidateValidPersonID(pWebUserID,pPersonID);
IF pPersonID IS NULL THEN
    SELECT '99:~Unique candidate link required';
    RETURN;
END IF;
IF pshifttype NOT IN( 'H','A','U') THEN //Holiday/Available/Unavailable
    SELECT '101:~invalid availability type';
    RETURN
END IF;
IF(TRIM(isnull(pdatefrom,'')) <> '' AND dfrom IS NULL)
OR(TRIM(isnull(pdateto,'')) <> '' AND dto IS NULL)
OR(TRIM(isnull(ptimefrom,'')) <> '' AND tfrom IS NULL)
OR(TRIM(isnull(ptimeto,'')) <> '' AND tto IS NULL) THEN
    SELECT '102:~Invalid date or time';
    RETURN
END IF;
IF dfrom IS NULL THEN
    SELECT '103:~Invalid date';
    RETURN
END IF;
IF dto < dfrom THEN
    SELECT '104:~Finish day appears to be before Start day';
    RETURN;
END IF;
SET dto=isnull(dto,dfrom);
IF tfrom IS NULL THEN
    SET tto=NULL
END IF;
IF tto IS NULL THEN
    SET tfrom=NULL
END IF;
while dfrom <= dto loop
    INSERT INTO tempshift(
tempshiftid,personid,shiftdate,timefrom,timeto,state) VALUES(
uniquekey(dfrom+tfrom),pPersonID,dfrom,tfrom,tto,pshifttype) ;
    SET dfrom=dfrom+1
```



```
END loop;
SELECT '0:~Success'
END
/* DOC
2016-12-01 PC Doc and test
2018-01-23 PC modify primary key seed to avoid duplicating on several events
on same day but different time
2018-05-17 MHS Fix and simplify tests. Allowed fromtime>totime for overnight
availability etc.
2018-07-24 PC include improved error handler
2018-10-10 MHS IW-456 multi-cand fix. For shift updates and deletes it only
affects them if they belong to the cand. For inserts it insists on unique or
explicit pPersonID.

pdatefrom dd/mm/yyyy : Shift date - use when adding or editing holiday,
availability or unavailability.
pdateto dd/mm/yyyy : End of date range - use when adding holiday,
availability or unavailability.
ptimefrom and
ptimeto hh:mm : use when editing or adding holiday, availability or
unavailability. Blank for whole days.
pshifttype: Use when adding A - availability, H - holiday or U -
unavailability.
IDCode: The tempShiftID - use when editing or deleting existing holiday,
availability or unavailability,
    or confirming or cancelling a shift. Blank otherwise.
pState: C for cancel shift. F for confirm shift. D for delete holiday,
availability or unavailability. Blank otherwise.

Inputs:
    pWebUserID
    pshifttype
    pdatefrom
    pdateto
    ptimefrom
    ptimeto
    IDCode
    pState

Outputs:
    pResult

*/

/* TEST
call NetTestSetup('');
delete from tempshift where personid='TEST';
select * from NetCandidateDiaryAdd(null,null,null,null,null,null,null,null);
```



```
expect 99*;
select * from
NetCandidateDiaryAdd('test.candidate',null,null,null,null,null,null,null);
expect 101*;
select * from
NetCandidateDiaryAdd('test.candidate','H',null,null,null,null,null,null);
expect 103*;
select * from
NetCandidateDiaryAdd('test.candidate','',null,null,null,null,null,'F');
expect 0*;
select * from
NetCandidateDiaryAdd('test.candidate','',null,null,null,null,null,'C');
expect 0*;
select * from
NetCandidateDiaryAdd('test.candidate','',null,null,null,null,null,'D');
expect 0*;
select * from
NetCandidateDiaryAdd('test.candidate','H',dateformat(now(*),'dd/mm/yyyy'),dateformat(now(*),'dd/mm/yyyy'),dateformat(now(*),'hh:mm:ss'),dateformat(dateadd(dd,1,now(*)),'hh:mm:ss'),null,null);
expect 0*;
select * from tempshift where personid='TEST';
expect State=H;
expect EOF;
*/
}
```

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

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

Last update: 2026/08/07 19:24

