



pears.NetCandidateDoTermsAndConditions



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

Original SQL

```
CREATE PROCEDURE "pears"."NetCandidateDoTermsAndConditions"( IN "pWebUserID"
CHAR(20),IN "pStage" CHAR(1),IN "pDocumentID" CHAR(20) DEFAULT NULL,IN
"pDivisionID" CHAR(20) DEFAULT NULL,IN "pDepartmentID" CHAR(2) DEFAULT
NULL,IN "pPersonID" CHAR(20) DEFAULT NULL )
RESULT( "pResult" long VARCHAR )
// IQXNet
BEGIN
  DECLARE "SourceDocID" CHAR(20);
  DECLARE "NewDocID" CHAR(20);
  SET "pPersonID" = "NetCandidateValidPersonID"("pWebUserID","pPersonID");
  IF "pPersonID" IS NULL THEN
    SELECT '97:~Unique candidate link required';
    RETURN
  END IF;
  IF "pStage" = 'A' AND NOT "pDocumentID" = any(SELECT "oledocumentid" FROM
"oledocument" WHERE "ownertype" = 'P' AND "ownerid" = "pPersonID") THEN
    SELECT '99:~Permission denied';
    RETURN
  END IF;
  IF "pStage" = '0' THEN
    SET "SourceDocID" = (SELECT top 1 "globaldocumentid"
FROM(SELECT top 1 "globaldocumentid",
IF EXISTS(SELECT * FROM "globaldocumentdetail" WHERE
"globaldocumentid" = "gd"."globaldocumentid" AND "doctype" = 'DIV' AND
"docid" = "pDivisionID") THEN 1 ELSE 0 endif AS "matchDiv",
IF EXISTS(SELECT * FROM "globaldocumentdetail" WHERE
"globaldocumentid" = "gd"."globaldocumentid" AND "doctype" = 'DEPT' AND
"docid" = "pDepartmentID") THEN 1 ELSE 0 endif AS "matchDept",
IF EXISTS(SELECT * FROM "globaldocumentdetail" WHERE
"globaldocumentid" = "gd"."globaldocumentid" AND "doctype" = 'DIV') THEN
ELSE 1 endif AS "globalDiv",
IF EXISTS(SELECT * FROM "globaldocumentdetail" WHERE
"globaldocumentid" = "gd"."globaldocumentid" AND "doctype" = 'DEPT') THEN
ELSE 1 endif AS "globalDept",
"matchDiv"+"matchDept" AS "score"
```



```
FROM "globaldocument" AS "gd"
WHERE "description" = 'WEBFETCHTERMS' AND("matchDiv" = 1 OR
"globalDiv" = 1) AND("matchDept" = 1 OR "globalDept" = 1) AND
"isnull"("defunct",0) = 0 AND "iqxnetuserclassid" = 'CANDIDATE'
ORDER BY "score" DESC) AS "docs" ORDER BY "globaldocumentid"
ASC);
IF "SourceDocID" IS NULL THEN
    SELECT '0:~No available T&Cs';
    RETURN
END IF;
SET "NewDocID" = "uniquekey"('');
IF "BlobstoreCopy"('G',"SourceDocID",'0',"NewDocID",'IQXWeb') <> 0 THEN
    SELECT '5:~Unable to generate T&Cs';
    RETURN
END IF;
INSERT INTO "oledocument"(
"oledocumentid","description","ownertype","ownerid","rawfile","fileextension
" ) SELECT "NewDocID",'T&Cs Offered '+'dateformat"(CURRENT
TIMESTAMP,'dd/mm/yyyy hh:nn'),'P',"pPersonID",1,"fileextension" FROM
"globaldocument" WHERE "globaldocumentid" = "SourceDocID";
UPDATE "blobstore" SET "publishtoweb" = 1 WHERE "class" = '0' AND "id" =
"NewDocID";
SELECT '0:+' "NewDocID" + '~Success';
RETURN
ELSE IF "pStage" = 'A' THEN
    UPDATE "oledocument" SET "description" = 'T&Cs Accepted
+' "dateformat"(CURRENT TIMESTAMP,'dd/mm/yyyy hh:nn') WHERE "oledocumentid" =
"pDocumentID";
    SELECT '0:+' "pDocumentID" + '~Success';
    RETURN
ELSE IF "pStage" = 'R' THEN
    DELETE FROM "phone" WHERE "who" = 'P' AND "whoid" = "pPersonID";
    DELETE FROM "tagvalue" WHERE "taglocation" = 'P' AND "id" =
"pPersonID";
    DELETE FROM "iqxnetuserlink" WHERE "personid" = "pPersonID";
    DELETE FROM "iqxnetuser" WHERE "iqxnetuserid" = "pWebUserID";
    DELETE FROM "blobstore" WHERE "class" = '0' AND "id" = any(SELECT
"oledocumentid" FROM "oledocument" WHERE "ownertype" = 'P' AND "ownerid" =
"pPersonID");
    DELETE FROM "oledocument" WHERE "ownertype" = 'P' AND "ownerid" =
"pPersonID";
    DELETE FROM "person" WHERE "personid" = "pPersonID";
    SELECT '0:~Success';
    RETURN
ELSE
    SELECT '3:~Invalid pStage';
    RETURN
END IF
```



```
END IF
END IF
END /* DOC
pStage: 0=Offer, A=Accept, R=Refuse
2018-07-24 PC include improved error handler
2018-10-22 MHS IW-456 multi-cand fix and tests
2018-12-12 GJ Removed pwebuserid check and replaced with pPersonID
2019-03-05 PC IW-873 add top 1 and order by on subqueries
2020-6-9 MHS added extra 'top 1' to ensure correct selection of specific
matching doc over general fallback
*/
/* TEST
call NetTestSetup('');
delete from blobstore where class='0' and id in (select oledocumentid from
oledocument where ownertype='P' and ownerid='TEST');
delete from oledocument where ownertype='P' and ownerid='TEST';
select count(*) from oledocument where ownertype='P' and ownerid='TEST';
expect 0;
CREATE OR REPLACE VARIABLE iqblobtemp long binary = 'abc';
select blobstorewrite('G','TESTTCGLOB',0,'WebTest');
expect 0;
insert into globaldocument
(globaldocumentid,description,iqxnetuserclassid,fileextension) on existing
update values ('TESTTCGLOB','WEBFETCHTERMS','CANDIDATE','pdf');
select pResult from NetCandidateDoTermsAndConditions('test.candidate','0');
expect 0:*;
select count(*) from oledocument where ownertype='P' and ownerid='TEST' and
description like '%offered%';
expect 1;
select pResult from
NetCandidateDoTermsAndConditions('test.candidate','A',(select first
oledocumentid from oledocument where ownertype='P' and ownerid='TEST'));
expect 0:*;
select count(*) from oledocument where ownertype='P' and ownerid='TEST' and
description like '%accepted%';
expect 1;
*/
GO

COMMENT TO PRESERVE FORMAT ON PROCEDURE
"pears"."NetCandidateDoTermsAndConditions" IS
{CREATE PROCEDURE pears."NetCandidateDoTermsAndConditions"(IN pWebUserID
CHAR(20), IN pStage CHAR(1), IN pDocumentID CHAR(20) DEFAULT NULL, IN
pDivisionID CHAR(20) DEFAULT NULL, IN pDepartmentID CHAR(2) DEFAULT NULL, IN
pPersonID CHAR(20) DEFAULT NULL )
RESULT(pResult long VARCHAR )
// IQXNet
BEGIN
```



```
DECLARE SourceDocID CHAR(20);
DECLARE NewDocID CHAR(20);
SET pPersonID=NetCandidateValidPersonID(pWebUserID,pPersonID);
IF pPersonID IS NULL THEN
    SELECT '97:~Unique candidate link required';
    RETURN;
END IF;
IF pStage = 'A' AND pDocumentID NOT IN (SELECT oledocumentid FROM
oledocument WHERE ownertype='P' AND ownerid=pPersonID) THEN
    SELECT '99:~Permission denied';
    RETURN;
END IF;

IF pStage='0' THEN
    SET SourceDocID = (SELECT top 1 globaldocumentid FROM
        (SELECT top 1 globaldocumentid,
            IF EXISTS(SELECT * FROM globaldocumentdetail WHERE
globaldocumentid=gd.globaldocumentid AND doctype='DIV' AND
docid=pDivisionID) THEN 1 ELSE 0 endif AS matchDiv,
            IF EXISTS(SELECT * FROM globaldocumentdetail WHERE
globaldocumentid=gd.globaldocumentid AND doctype='DEPT' AND
docid=pDepartmentID) THEN 1 ELSE 0 endif AS matchDept,
            IF EXISTS(SELECT * FROM globaldocumentdetail WHERE
globaldocumentid=gd.globaldocumentid AND doctype='DIV') THEN 0 ELSE 1 endif
AS globalDiv,
            IF EXISTS(SELECT * FROM globaldocumentdetail WHERE
globaldocumentid=gd.globaldocumentid AND doctype='DEPT') THEN 0 ELSE 1 endif
AS globalDept,
            matchDiv + matchDept AS score
        FROM globaldocument gd
        WHERE
            description='WEBFETCHTERMS' AND (matchDiv=1 OR globalDiv=1) AND
(matchDept=1 OR globalDept=1) AND isnull(defunct,0)=0 AND
iqxnetuserclassid='CANDIDATE'
        ORDER BY score DESC ) AS docs ORDER BY globaldocumentid);
    IF SourceDocID IS NULL THEN
        SELECT '0:~No available T&Cs';
        RETURN;
    END IF;
    SET NewDocID=uniquekey('');
    IF BlobstoreCopy('G',SourceDocID,'0',NewDocID,'IQXWeb')<>0 THEN
        SELECT '5:~Unable to generate T&Cs';
        RETURN;
    END IF;
    INSERT INTO
oledocument(oledocumentid,description,ownertype,ownerid,rawfile,fileextensio
n) SELECT NewDocID,'T&Cs Offered '+dateformat(CURRENT_TIMESTAMP, 'dd/mm/yyyy
hh:nn'),'P',pPersonID,1,fileextension FROM globaldocument WHERE
```



```
globaldocumentid=SourceDocID;
  UPDATE blobstore SET publishtoweb=1 WHERE class='0' AND id=NewDocID;
  SELECT '0:'+NewDocID+'~Success';
  RETURN;
ELSE IF pStage='A' THEN
  UPDATE oledocument SET description='T&Cs Accepted '+dateformat(CURRENT
TIMESTAMP, 'dd/mm/yyyy hh:nn') WHERE oledocumentid=pDocumentID;
  SELECT '0:'+pDocumentID+'~Success';
  RETURN;
ELSE IF pStage='R' THEN
  DELETE FROM phone WHERE who='P' AND whoid=pPersonID;
  DELETE FROM tagvalue WHERE taglocation='P' AND id=pPersonID;
  DELETE FROM iqxnetuserlink WHERE personid=pPersonID;
  DELETE FROM iqxnetuser WHERE iqxnetuserid=pWebUserID;
  DELETE FROM blobstore WHERE class='0' AND id IN (SELECT oledocumentid FROM
oledocument WHERE ownertype='P' AND ownerid=pPersonID);
  DELETE FROM oledocument WHERE ownertype='P' AND ownerid=pPersonID;
  DELETE FROM person WHERE personid=pPersonID;
  SELECT '0:~Success';
  RETURN;
ELSE
  SELECT '3:~Invalid pStage';
  RETURN;
END IF;
END IF;
END IF;
END
```

```
/* DOC
```

```
pStage: 0=Offer, A=Accept, R=Refuse
```

```
2018-07-24 PC include improved error handler
```

```
2018-10-22 MHS IW-456 multi-cand fix and tests
```

```
2018-12-12 GJ Removed pwebuserid check and replaced with pPersonID
```

```
2019-03-05 PC IW-873 add top 1 and order by on subqueries
```

```
2020-6-9 MHS added extra 'top 1' to ensure correct selection of specific
matching doc over general fallback
```

```
*/
```

```
/* TEST
```

```
call NetTestSetup('');
```

```
delete from blobstore where class='0' and id in (select oledocumentid from
oledocument where ownertype='P' and ownerid='TEST');
```

```
delete from oledocument where ownertype='P' and ownerid='TEST';
```

```
select count(*) from oledocument where ownertype='P' and ownerid='TEST';
```

```
expect 0;
```

```
CREATE OR REPLACE VARIABLE iqblobtemp long binary = 'abc';
```

```
select blobstorewrite('G','TESTTCGLOB',0,'WebTest');
```

```
expect 0;
```



```
insert into globaldocument
(globaldocumentid,description,iqxnetuserclassid,fileextension) on existing
update values ('TESTTCGLOB','WEBFETCHTERMS','CANDIDATE','pdf');
select pResult from NetCandidateDoTermsAndConditions('test.candidate','0');
expect 0:*;
select count(*) from oledocument where ownertype='P' and ownerid='TEST' and
description like '%offered%';
expect 1;
select pResult from
NetCandidateDoTermsAndConditions('test.candidate','A',(select first
oledocumentid from oledocument where ownertype='P' and ownerid='TEST'));
expect 0:*;
select count(*) from oledocument where ownertype='P' and ownerid='TEST' and
description like '%accepted%';
expect 1;
*/
}
```

From:

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

Permanent link:

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

Last update: 2026/08/07 19:24

