



pears.MergeEmployments



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

Original SQL

```
CREATE PROCEDURE "pears"."MergeEmployments"(  
  /* Application Maintained Function / Procedure - DO NOT EDIT*/  
  IN "ssourceid" CHAR(20), IN "stargetid" CHAR(20) )  
BEGIN  
  DECLARE "snote" CHAR(100);  
  DECLARE "tnote" CHAR(100);  
  DECLARE "idum" SMALLINT;  
  DECLARE "snetid" CHAR(20);  
  SELECT "note" INTO "snote" FROM "employment" WHERE "employmentid" =  
"ssourceid";  
  SELECT "note" INTO "tnote" FROM "employment" WHERE "employmentid" =  
"stargetid";  
  IF "trim"("snote") <> '' THEN  
    UPDATE "employment" SET "note" = "trim"("string"("tnote",' ')) ||  
"string"("snote") WHERE "employmentid" = "stargetid"  
  END IF;  
  UPDATE "contactevent" SET "employmentid" = "stargetid" WHERE  
"employmentid" = "ssourceid";  
  UPDATE "diary" SET "employmentid" = "stargetid" WHERE "employmentid" =  
"ssourceid";  
  UPDATE "placement" SET "employmentid" = "stargetid" WHERE "employmentid" =  
"ssourceid";  
  BEGIN  
    UPDATE "tempslot" SET "employmentid" = "stargetid" WHERE "employmentid"  
= "ssourceid"  
  exception  
    WHEN others THEN SET "idum" = 0  
  END;  
  UPDATE "vacancy" SET "employmentid" = "stargetid" WHERE "employmentid" =  
"ssourceid";  
  UPDATE "etip" SET "employmentid" = "stargetid" WHERE "employmentid" =  
"ssourceid";  
  UPDATE "companyaccount" SET "accountscontact" = "stargetid" WHERE  
"accountscontact" = "ssourceid";  
  UPDATE "companyaccount" SET "timesheetcontact" = "stargetid" WHERE
```



```
"accountscontact" = "ssourceid";
  UPDATE "CompanySDS" SET "employmentid" = "targetid" WHERE "employmentid"
= "ssourceid";
  UPDATE "ReferenceRequest" SET "employmentid" = "targetid" WHERE
"employmentid" = "ssourceid";
  UPDATE "VacancyRoleAllocation" AS "t" SET "employmentid" = "targetid"
WHERE "employmentid" = "ssourceid";
  CALL "mergephones"('CP',"ssourceid","targetid");
  CALL "mergequestionnaire"('E',"ssourceid","targetid");
  IF EXISTS(SELECT * FROM "iqxnetuserlink" WHERE "employmentid" =
"targetid") THEN
    SELECT FIRST "iqxnetuserid" INTO "snetid" FROM "iqxnetuserlink" WHERE
"employmentid" = "ssourceid";
    DELETE FROM "iqxnetuserlink" WHERE "employmentid" = "ssourceid";
    DELETE FROM "iqxnetuser" WHERE "iqxnetuserid" = "snetid" AND NOT
EXISTS(SELECT * FROM "iqxnetuserlink" WHERE "iqxnetuserid" =
"iqxnetuser"."iqxnetuserid")
  ELSE
    UPDATE "iqxnetuserlink" SET "employmentid" = "targetid" WHERE
"employmentid" = "ssourceid"
  END IF;
  DELETE FROM "employment" WHERE "employmentid" = "ssourceid"
END
GO
```

```
COMMENT TO PRESERVE FORMAT ON PROCEDURE "pears"."MergeEmployments" IS
{CREATE PROCEDURE MergeEmployments
```

```
/* Application Maintained Function / Procedure - DO NOT EDIT*/
```

```
(
  IN ssourceid CHAR(20),IN targetid CHAR(20) )
BEGIN
  DECLARE snote CHAR(100);
  DECLARE tnote CHAR(100);
  DECLARE idum SMALLINT;
  DECLARE snetid CHAR(20);
  SELECT note INTO snote FROM employment WHERE employmentid = ssourceid;
  SELECT note INTO tnote FROM employment WHERE employmentid = targetid;
  IF TRIM(snote) <> '' THEN
    UPDATE employment SET note = TRIM(string(tnote,' ')) || string(snote)
WHERE employmentid = targetid
  END IF;
  UPDATE contactevent SET employmentid = targetid WHERE employmentid =
ssourceid;
  UPDATE diary SET employmentid = targetid WHERE employmentid = ssourceid;
  UPDATE placement SET employmentid = targetid WHERE employmentid =
ssourceid;
```



```
BEGIN
  UPDATE tempslot SET employmentid = targetid WHERE employmentid =
ssourceid
  exception
    WHEN others THEN SET idum = 0
END;
UPDATE vacancy SET employmentid = targetid WHERE employmentid =
ssourceid;
UPDATE etip SET employmentid = targetid WHERE employmentid = ssourceid;
UPDATE companyaccount SET accountscontact = targetid WHERE
accountscontact = ssourceid;
UPDATE companyaccount SET timesheetcontact = targetid WHERE
accountscontact = ssourceid;
UPDATE CompanySDS SET employmentid = targetid WHERE employmentid =
ssourceid;
UPDATE ReferenceRequest SET employmentid = targetid WHERE employmentid =
ssourceid;
UPDATE VacancyRoleAllocation t SET employmentid = targetid WHERE
employmentid = ssourceid ;
CALL mergephones('CP',ssourceid,targetid);
CALL mergequestionnaire('E',ssourceid,targetid);
IF EXISTS(SELECT * FROM iqxnetuserlink WHERE employmentid = targetid)
THEN
  SELECT FIRST iqxnetuserid INTO snetid FROM iqxnetuserlink WHERE
employmentid = ssourceid;
  DELETE FROM iqxnetuserlink WHERE employmentid = ssourceid;
  DELETE FROM iqxnetuser WHERE iqxnetuserid = snetid AND NOT EXISTS(SELECT
* FROM iqxnetuserlink WHERE iqxnetuserid = iqxnetuser.ixxnetuserid)
ELSE
  UPDATE iqxnetuserlink SET employmentid = targetid WHERE employmentid =
ssourceid
END IF;
DELETE FROM employment WHERE employmentid = ssourceid
END
}
```

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

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

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

