



pears.AWRCheck



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

Original SQL

```
CREATE PROCEDURE "pears"."AWRCheck"( IN "BookOrConfirmorPlaceorTSflag"
CHAR(1),IN "placeid" CHAR(20),IN @speciality CHAR(50),IN @grade CHAR(4),IN
"shiftid" CHAR(20),IN "WeekNo" INTEGER )
RESULT( "DoNothingflag" CHAR(1),"AskOtherAgencyWorkflag"
CHAR(1),"ShowAWRDetailsflag" CHAR(1),"AWRJobMasterID" CHAR(20),"UserMessage"
long VARCHAR,"LaterExpansionStuff" long VARCHAR )
BEGIN
  -- B: Shift booking
  -- C: Shift client confirm
  -- T: shift temp confirm
  -- S: At timesheet, when shift change of grade or speciality
  -- P: Placement
  -- L: At timesheet, when placement
  DECLARE "WeeksToWarn" INTEGER;
  DECLARE "WeeksToWarnTimesheet" INTEGER;
  DECLARE "WeeksToWarnBooking" INTEGER;
  DECLARE "DateToStartChecking" DATE;
  DECLARE "newid" CHAR(20);
  DECLARE "persid" CHAR(20);
  DECLARE "vacid" CHAR(20);
  DECLARE "jobmid" CHAR(20);
  DECLARE "AWRStat" SMALLINT;
  DECLARE "JAWRStat" SMALLINT;
  DECLARE "UMess" CHAR(2000);
  DECLARE "AWRLink" CHAR(20);
  DECLARE "coid" CHAR(20);
  DECLARE "LE" CHAR(50);
  -- this could be a question or jobtitle or something else
  DECLARE "LinkingQuestion" CHAR(50);
  SET "AWRLink" = NULL; -- set variables here
  -- set variables here
  SET "WeeksToWarnTimesheet" = 11;
  SET "WeeksToWarnBooking" = 11;
  SET "DateToStartChecking" = '2011-10-03';
  IF "varexists"('AWRStartPeriod') = 1 THEN
```



```
SET "DateToStartChecking" = "weekmonthenddate"("AWRStartPeriod",'W')
END IF;
-- set variables here
-- set variables here
-- AWRJobMaster always insert personid and vacancyid
--
-- check start date
IF "datediff"("day",CURRENT DATE,"DateToStartChecking") >= 0 THEN
    SELECT 'Y','Y','Y','','','';
    RETURN
END IF;
SET "Umess" = '';
IF "BookOrConfirmorPlaceorTSflag" = 'B' THEN
    SELECT "personid","vacancyid" INTO "persid","vacid" FROM "tempshift"
WHERE "tempshiftid" = "shiftid";
    IF EXISTS(SELECT * FROM "person" WHERE "personid" = "persid" AND
"isnull"("directlyemployed",0) >= 1) THEN
        SELECT 'Y','Y','Y','','','';
        RETURN
    END IF;
-- rules
-- dont ask again if already asked for booked ( or worked ) shift in
this week or untimesheeted = role with no place id
    IF NOT EXISTS(SELECT * FROM "tempshift" WHERE "personid" = "persid" AND
"vacancyid" = "vacid" AND "tempshiftid" <> "shiftid" AND "state" IN(
'P','B','W' ) AND "weekcontaining"("shiftdate") = "WeekNO") AND NOT
EXISTS(SELECT * FROM "AWRJobMaster" WHERE "personid" = "persid" AND
"vacancyid" = "vacid" AND "placementid" IS NULL) THEN
        SELECT "AWRStatus" INTO "AWRstat" FROM "AWRVacancy" WHERE "vacancyid"
= "vacid";
        SELECT FIRST "AWRStatus","AWRjobmasterid" INTO "JAWRstat","jobmid"
FROM "AWRjobmaster" WHERE "placementid" =
"AWRPlacement"("vacid","persid","shiftid");
        -- speciality and grade
        IF "JAWRstat" IS NULL THEN
            SET "JAWRstat" = 4
        ELSE
            IF "AWRWeeks"("jobmid","WeekNo",NULL,NULL,NULL,NULL) =
"WeeksToWarnBooking" THEN
                SET "Umess" = "string('This temp will qualify in ',12-
"WeeksToWarnBooking",' week(s) so the AWR details must be checked for Rates,
Holidays, Linked Placements and other work.')"
            END IF END IF;
-- 1 not known 2 applies 3 not
CASE "JAWRstat"
WHEN 1 THEN
    -- need to check work so far here
    IF "AWRGap"("jobmid","WeekNo",NULL,NULL,NULL,NULL) >= 6 THEN
```



```
        SELECT 'N','Y','Y',"jobmid",'',''
    ELSE
        SELECT 'N','N','Y',"jobmid","UMess",' '
    END IF WHEN 2 THEN
    IF "AWRGap"("jobmid","WeekNo",NULL,NULL,NULL,NULL) >= 6 THEN
        SELECT 'N','Y','N',"jobmid",'',''
    ELSE
        SELECT 'N','N','N',"jobmid","UMess",' '
    END IF WHEN 3 THEN
    SELECT 'Y','Y','Y',' ',' ',' '
WHEN 4 THEN
    -- no record check vac might have changed
    IF "AWRstat" = 3 THEN
        SELECT 'Y','Y','Y',' ',' ',' '
    ELSE
        SET "newid" = "uniquekey"("shiftid");
        SELECT
"getquestanswer"("string"('V',"vacancy"."departmentid"),'AWR',"vacancy"."vac
ancyid"),"LECode","company"."companyid" INTO "LinkingQuestion","LE","coid"
FROM "tempshift" KEY JOIN "vacancy" KEY JOIN "employment" KEY JOIN "company"
KEY JOIN "AWRCompany" WHERE "tempshiftid" = "shiftid";
        IF "LE" IS NOT NULL THEN
            SELECT FIRST "AWRJobMasterID" INTO "AWRLink" FROM "AWRJobMaster"
KEY JOIN "placement" KEY JOIN("vacancy","employment" AS "e")
            -- find "master" linked record in case some already linked
            WHERE "AWRJobMasterID" = "AWRLinkCode"
            -- LE code check
            AND EXISTS(SELECT "CompanyID" FROM "AWRCompany" WHERE "LECode"
= "LE" AND "companyid" = "e"."companyid")
            -- job title or question check
            AND
"getquestanswer"("string"('V',"vacancy"."departmentid"),'AWR',"vacancy"."vac
ancyid") = "LinkingQuestion"
            AND "LinkingQuestion" <> ''
            AND "e"."personid" = "persid"
        ELSE
            SELECT FIRST "AWRJobMasterID" INTO "AWRLink" FROM "AWRJobMaster"
KEY JOIN "placement" KEY JOIN("vacancy","employment" AS "e")
            -- find "master" linked record in case some already linked
            WHERE "AWRJobMasterID" = "AWRLinkCode"
            -- LE code check
            AND "e"."companyid" = "coid"
            -- job title or question check
            AND "isnull"("e"."position","vacancy"."position") =
"LinkingQuestion"
            AND "LinkingQuestion" <> ''
            AND "e"."personid" = "persid"
        END IF;
```



```
INSERT INTO "AWRJobmaster"(  
"awrjobmasterid","placementid","grade","speciality","vacancyid","personid","  
AWRLinkCode" ) VALUES(  
"newid","AWRPlacement"("vacid","persid","shiftid"),@grade,@speciality,"vacid"  
,"persid","AWRLink" ) ;  
SELECT 'N','Y','Y',"newid",'',''  
END IF  
END CASE  
ELSE SELECT 'Y','Y','Y',"newid","string('',' ' x'),''  
END IF END IF;  
IF "BookOrConfirmorPlaceorTSflag" = 'C' THEN  
SELECT 'Y','Y','Y','','',''  
END IF; -- testing code select 'N','Y','Y',(select first awrjobmasterid  
from awrjobmaster where vacancyid is not null and personid is not null),'You  
must ask about other agency work and must check the AWR details client  
confirm',''  
IF "BookOrConfirmorPlaceorTSflag" = 'P' THEN  
SELECT "employment"."personid","placement"."vacancyid" INTO  
"persid","vacid" FROM "placement" KEY JOIN "employment" WHERE "placementid"  
= "placeid";  
IF EXISTS(SELECT * FROM "person" WHERE "personid" = "persid" AND  
"isnull"("directlyemployed",0) >= 1) THEN  
SELECT 'Y','Y','Y','','','';  
RETURN  
END IF;  
SELECT FIRST "awrvacancy"."AWRStatus" INTO "AWRstat" FROM "AWRVacancy"  
JOIN "placement" ON "AWRVacancy"."vacancyid" = "placement"."vacancyid" WHERE  
"placement"."placementid" = "placeid";  
IF "AWRStat" IS NULL THEN  
SET "Umess" = 'No Vacancy AWR has been set up. You will be asked to  
fill in AWR details for the Placement, later you should also update the  
Vacancy.';  
SET "AWRStat" = 1  
END IF;  
IF "AWRStat" IN( 1,2 ) THEN  
SELECT  
"getquestanswer"("string"('V',"vacancy"."departmentid"),'AWR',"vacancy"."vac  
ancyid"),"LECode","company"."companyid" INTO "LinkingQuestion","LE","coid"  
FROM "placement" KEY JOIN "employment" KEY JOIN "company" KEY JOIN  
"AWRCompany","placement" KEY JOIN "vacancy" WHERE "placement"."placementid"  
= "placeid";  
IF "LE" IS NOT NULL THEN  
SELECT FIRST "AWRJobMasterID" INTO "AWRLink" FROM "AWRJobMaster" KEY  
JOIN "placement" KEY JOIN("vacancy","employment" AS "e")  
-- find "master" linked record in case some already linked  
WHERE "AWRJobMasterID" = "AWRLinkCode"  
-- LE code check  
AND EXISTS(SELECT "CompanyID" FROM "AWRCompany" WHERE "LECode" =
```



```
"LE" AND "companyid" = "e"."companyid")
    -- job title or question check
    AND
"getquestanswer"("string"('V',"vacancy"."departmentid"),'AWR',"vacancy"."vacancyid") = "LinkingQuestion"
    AND "LinkingQuestion" <> ''
    AND "e"."personid" = "persid"
ELSE
    SELECT FIRST "AWRJobMasterID" INTO "AWRLink" FROM "AWRJobMaster" KEY
JOIN "placement" KEY JOIN("vacancy","employment" AS "e")
    -- find "master" linked record in case some already linked
    WHERE "AWRJobMasterID" = "AWRLinkCode"
    -- LE code check
    AND "e"."companyid" = "coid"
    -- job title or question check
    AND "isnull"("e"."position","vacancy"."position") =
"LinkingQuestion"
    AND "LinkingQuestion" <> ''
    AND "e"."personid" = "persid"
END IF END IF;
CASE "AWRStat"
WHEN 1 THEN
    SET "newid" = "uniquekey"("placeid");
    INSERT INTO "AWRJobmaster"(
"awrjobmasterid","placementid","grade","speciality","vacancyid","personid","
AWRLinkCode" ) VALUES(
"newid","placeid",@grade,@speciality,"vacid","persid","AWRLink" ) ;
    SELECT 'N','Y','Y',"newid","UMess",'
WHEN 2 THEN
    SET "newid" = "uniquekey"("placeid");
    INSERT INTO "AWRJobmaster"(
"awrjobmasterid","placementid","grade","speciality","vacancyid","personid","
AWRLinkCode" ) VALUES(
"newid","placeid",@grade,@speciality,"vacid","persid","AWRLink" ) ;
    SELECT 'N','Y','N',"newid",'',''
WHEN 3 THEN
    SELECT 'Y','Y','Y',' ',' ',' '
END CASE
END IF;
IF "BookOrConfirmorPlaceorTSflag" = 'T' THEN
    SELECT 'Y','Y','Y',' ',' ',' '
END IF; -- testing code    select 'N','Y','Y',(select first awrjobmasterid
from awrjobmaster where vacancyid is not null and personid is not null),'You
must ask about other agency work and must check the AWR details temp
confirm',''
IF "BookOrConfirmorPlaceorTSflag" = 'S' THEN
    SELECT 'Y','Y','Y',' ',' ',' '
END IF; -- testing code    select 'N','N','Y',(select first awrjobmasterid
```



```
from awrjobmaster where vacancyid is not null and personid is not null),'You
must ask about other agency work and must check the AWR details shift
timesheet',''
  IF "BookOrConfirmorPlaceorTSflag" = 'L' THEN
    IF "placeid" IS NULL THEN
      SELECT 'N','N','N',NULL,"string"('For AWR compatability if this
timesheet relates to a vacancy you must create a placement before entering a
timesheet.','char"(13),"char"(10),' Go back, create the placement and try
again.'),'';
      RETURN
    END IF;
    SELECT "employment"."personid","placement"."vacancyid" INTO
"persid","vacid" FROM "placement" KEY JOIN "employment" WHERE
"placement"."placementid" = "placeid";
    IF EXISTS(SELECT * FROM "person" WHERE "personid" = "persid" AND
"isnull"("directlyemployed",0) >= 1) THEN
      SELECT 'Y','Y','Y',' ',' ',' ';
      RETURN
    END IF;
    SELECT FIRST "AWRVacancy"."AWRStatus" INTO "AWRstat" FROM "AWRVacancy"
JOIN "placement" ON "AWRVacancy"."vacancyid" = "placement"."vacancyid" WHERE
"placement"."placementid" = "placeid";
    SELECT FIRST "AWRStatus","AWRjobmasterid" INTO "JAWRstat","jobmid" FROM
"AWRjobmaster" WHERE "placementid" = "placeid";
    -- speciality and grade
    IF "JAWRstat" IS NULL THEN
      SET "JAWRstat" = 4
    ELSE
      IF "AWRWeeks"("jobmid","WeekNo",NULL,NULL,NULL,NULL) =
"WeeksToWarnTimesheet" THEN
        SET "Umess" = "string"('This temp will qualify in ',12-
"WeeksToWarnTimesheet",' week(s). Check the AWR Details for what needs to be
done. Check for placements that should be linked or for work for other
Agencies.')
      END IF END IF;
    -- 1 not known 2 applies 3 not
    CASE "JAWRstat"
    WHEN 1 THEN
      -- need to check work so far here
      IF "AWRGap"("jobmid","WeekNo",NULL,NULL,NULL,NULL) >= 6 THEN
        SELECT 'N','Y','Y',"jobmid",' ',' '
      ELSE
        SELECT 'N','N','Y',"jobmid","UMess",' '
      END IF WHEN 2 THEN
      IF "AWRGap"("jobmid","WeekNo",NULL,NULL,NULL,NULL) >= 6 THEN
        SELECT 'N','Y','N',"jobmid",' ',' '
      ELSE
        SELECT 'N','N','N',"jobmid","UMess",' '
      END IF
```



```
END IF WHEN 3 THEN
SELECT 'Y','Y','Y','','',''
WHEN 4 THEN
-- no record check vac might have changed
IF "AWRstat" = 3 THEN
SELECT 'Y','Y','Y','','',''
ELSE
SET "newid" = "uniquekey"("placeid");
SELECT
"getquestanswer"("string"('V',"vacancy"."departmentid"),'AWR',"vacancy"."vac
ancyid"),"LECode","company"."companyid" INTO "LinkingQuestion","LE","coid"
FROM "placement" KEY JOIN "employment" KEY JOIN "company" KEY JOIN
"AWRCompany","placement" KEY JOIN "vacancy" WHERE "placement"."placementid"
= "placeid";
IF "LE" IS NOT NULL THEN
SELECT FIRST "AWRJobMasterID" INTO "AWRLink" FROM "AWRJobMaster"
KEY JOIN "placement" KEY JOIN("vacancy","employment" AS "e")
-- find "master" linked record in case some already linked
WHERE "AWRJobMasterID" = "AWRLinkCode"
-- LE code check
AND EXISTS(SELECT "CompanyID" FROM "AWRCompany" WHERE "LECode" =
"LE" AND "companyid" = "e"."companyid")
-- job title or question check
AND "isnull"("e"."position","vacancy"."position") =
"LinkingQuestion"
AND "LinkingQuestion" <> ''
AND "e"."personid" = "persid"
ELSE
SELECT FIRST "AWRJobMasterID" INTO "AWRLink" FROM "AWRJobMaster"
KEY JOIN "placement" KEY JOIN("vacancy","employment" AS "e")
-- find "master" linked record in case some already linked
WHERE "AWRJobMasterID" = "AWRLinkCode"
-- LE code check
AND "e"."companyid" = "coid"
-- job title or question check
AND
"getquestanswer"("string"('V',"vacancy"."departmentid"),'AWR',"vacancy"."vac
ancyid") = "LinkingQuestion"
AND "LinkingQuestion" <> ''
AND "e"."personid" = "persid"
END IF;
INSERT INTO "AWRJobmaster"(
"awrjobmasterid","placementid","grade","speciality","vacancyid","personid","
AWRLinkCode" ) VALUES(
"newid","placeid",@grade,@speciality,"vacid","persid","AWRLink" ) ;
SELECT 'N','Y','Y',"newid",'',''
END IF
END CASE
```



```
END IF
END
GO

COMMENT TO PRESERVE FORMAT ON PROCEDURE "pears"."AWRCheck" IS
{CREATE PROCEDURE pears."AWRCheck"( IN "BookOrConfirmorPlaceorTSflag"
CHAR(1),IN "placeid" CHAR(20),IN @speciality CHAR(50),IN @grade CHAR(4),IN
"shiftid" CHAR(20),IN "WeekNo" INTEGER )
RESULT(DoNothingflag CHAR(1),AskOtherAgencyWorkflag
CHAR(1),ShowAWRDetailsflag CHAR(1),AWRJobMasterID CHAR(20),UserMessage long
VARCHAR,LaterExpansionStuff long VARCHAR)
BEGIN
  -- B: Shift booking
  -- C: Shift client confirm
  -- T: shift temp confirm
  -- S: At timesheet, when shift change of grade or speciality
  -- P: Placement
  -- L: At timesheet, when placement
  DECLARE "WeeksToWarn" INTEGER;
  DECLARE "WeeksToWarnTimesheet" INTEGER;
  DECLARE "WeeksToWarnBooking" INTEGER;
  DECLARE "DateToStartChecking" DATE;
  DECLARE "newid" CHAR(20);
  DECLARE "persid" CHAR(20);
  DECLARE "vacid" CHAR(20);
  DECLARE "jobmid" CHAR(20);
  DECLARE "AWRStat" SMALLINT;
  DECLARE "JAWRStat" SMALLINT;
  DECLARE "UMess" CHAR(2000);
  DECLARE "AWRLink" CHAR(20);
  DECLARE "coid" CHAR(20);
  DECLARE "LE" CHAR(50);
  -- this could be a question or jobtitle or something else
  DECLARE "LinkingQuestion" CHAR(50);
  SET "AWRLink" = NULL; -- set variables here
  -- set variables here
  SET "WeeksToWarnTimesheet" = 11;
  SET "WeeksToWarnBooking" = 11;
  SET "DateToStartChecking" = '2011-10-03';
  IF "varexists"('AWRStartPeriod') = 1 THEN
    SET "DateToStartChecking" = "weekmonthenddate"("AWRStartPeriod",'W')
  END IF;
  -- set variables here
  -- set variables here
  -- AWRJobMaster always insert personid and vacancyid
  --
  -- check start date
  IF "datediff"("day",CURRENT DATE,"DateToStartChecking") >= 0 THEN
```




```
SELECT 'Y','Y','Y',' ',' ',' ';
RETURN
END IF;
SET "Umess" = ' ';
IF "BookOrConfirmorPlaceorTSflag" = 'B' THEN
  SELECT "personid","vacancyid" INTO "persid","vacid" FROM "tempshift"
WHERE "tempshiftid" = "shiftid";
  IF EXISTS(SELECT * FROM "person" WHERE "personid" = "persid" AND
"isnull"("directlyemployed",0) >= 1) THEN
    SELECT 'Y','Y','Y',' ',' ',' ';
    RETURN
  END IF;
  -- rules
  -- dont ask again if already asked for booked ( or worked ) shift in
this week or untimesheeted = role with no place id
  IF NOT EXISTS(SELECT * FROM "tempshift" WHERE "personid" = "persid" AND
"vacancyid" = "vacid" AND "tempshiftid" <> "shiftid" AND "state" IN(
'P','B','W' ) AND "weekcontaining"("shiftdate") = "WeekNO") AND NOT
EXISTS(SELECT * FROM "AWRJobMaster" WHERE "personid" = "persid" AND
"vacancyid" = "vacid" AND "placementid" IS NULL) THEN
    SELECT "AWRStatus" INTO "AWRstat" FROM "AWRVacancy" WHERE "vacancyid"
= "vacid";
    SELECT FIRST "AWRStatus","AWRjobmasterid" INTO "JAWRstat","jobmid"
FROM "AWRjobmaster" WHERE "placementid" =
"AWRPlacement"("vacid","persid","shiftid");
    -- speciality and grade
    IF "JAWRstat" IS NULL THEN
      SET "JAWRstat" = 4
    ELSE
      IF "AWRWeeks"("jobmid","WeekNo",NULL,NULL,NULL,NULL) =
"WeeksToWarnBooking" THEN
        SET "Umess" = "string('This temp will qualify in ',12-
"WeeksToWarnBooking",' week(s) so the AWR details must be checked for Rates,
Holidays, Linked Placements and other work.')"
      END IF END IF;
      -- 1 not known 2 applies 3 not
CASE "JAWRstat"
WHEN 1 THEN
  -- need to check work so far here
  IF "AWRGap"("jobmid","WeekNo",NULL,NULL,NULL,NULL) >= 6 THEN
    SELECT 'N','Y','Y',"jobmid",' ',' '
  ELSE
    SELECT 'N','N','Y',"jobmid","UMess",' '
  END IF WHEN 2 THEN
  IF "AWRGap"("jobmid","WeekNo",NULL,NULL,NULL,NULL) >= 6 THEN
    SELECT 'N','Y','N',"jobmid",' ',' '
  ELSE
    SELECT 'N','N','N',"jobmid","UMess",' '
  END IF
```



```
END IF WHEN 3 THEN
SELECT 'Y','Y','Y',' ',' ',' '
WHEN 4 THEN
-- no record check vac might have changed
IF "AWRstat" = 3 THEN
SELECT 'Y','Y','Y',' ',' ',' '
ELSE
SET "newid" = "uniquekey"("shiftid");
SELECT
"getquestanswer"("string"('V',"vacancy"."departmentid"),'AWR',"vacancy"."vac
ancyid"),"LECode","company"."companyid" INTO "LinkingQuestion","LE","coid"
FROM "tempshift" KEY JOIN "vacancy" KEY JOIN "employment" KEY JOIN "company"
KEY JOIN "AWRCompany" WHERE "tempshiftid" = "shiftid";
IF "LE" IS NOT NULL THEN
SELECT FIRST "AWRJobMasterID" INTO "AWRLink" FROM "AWRJobMaster"
KEY JOIN "placement" KEY JOIN("vacancy","employment" AS "e")
-- find "master" linked record in case some already linked
WHERE "AWRJobMasterID" = "AWRLinkCode"
-- LE code check
AND EXISTS(SELECT "CompanyID" FROM "AWRCompany" WHERE "LECode"
= "LE" AND "companyid" = "e"."companyid")
-- job title or question check
AND
"getquestanswer"("string"('V',"vacancy"."departmentid"),'AWR',"vacancy"."vac
ancyid") = "LinkingQuestion"
AND "LinkingQuestion" <> ''
AND "e"."personid" = "persid"
ELSE
SELECT FIRST "AWRJobMasterID" INTO "AWRLink" FROM "AWRJobMaster"
KEY JOIN "placement" KEY JOIN("vacancy","employment" AS "e")
-- find "master" linked record in case some already linked
WHERE "AWRJobMasterID" = "AWRLinkCode"
-- LE code check
AND "e"."companyid" = "coid"
-- job title or question check
AND "isnull"("e"."position","vacancy"."position") =
"LinkingQuestion"
AND "LinkingQuestion" <> ''
AND "e"."personid" = "persid"
END IF;
INSERT INTO "AWRJobmaster"(
"awrjobmasterid","placementid","grade","speciality","vacancyid","personid","
AWRLinkCode" ) VALUES(
"newid","AWRPlacement"("vacid","persid","shiftid"),@grade,@speciality,"vacid
","persid","AWRLink" ) ;
SELECT 'N','Y','Y',"newid",' ',' '
END IF
END CASE
```



```
ELSE SELECT 'Y','Y','Y',"newid","string('',' ' x')',''
END IF END IF;
IF "BookOrConfirmorPlaceorTSflag" = 'C' THEN
  SELECT 'Y','Y','Y',' ',' ',' '
END IF; -- testing code select 'N','Y','Y',(select first awrjobmasterid
from awrjobmaster where vacancyid is not null and personid is not null),'You
must ask about other agency work and must check the AWR details client
confirm',''
IF "BookOrConfirmorPlaceorTSflag" = 'P' THEN
  SELECT "employment"."personid","placement"."vacancyid" INTO
"persid","vacid" FROM "placement" KEY JOIN "employment" WHERE "placementid"
= "placeid";
  IF EXISTS(SELECT * FROM "person" WHERE "personid" = "persid" AND
"isnull"("directlyemployed",0) >= 1) THEN
    SELECT 'Y','Y','Y',' ',' ',' '
    RETURN
  END IF;
  SELECT FIRST "awrvacancy"."AWRStatus" INTO "AWRstat" FROM "AWRVacancy"
JOIN "placement" ON "AWRVacancy"."vacancyid" = "placement"."vacancyid" WHERE
"placement"."placementid" = "placeid";
  IF "AWRstat" IS NULL THEN
    SET "Umess" = 'No Vacancy AWR has been set up. You will be asked to
fill in AWR details for the Placement, later you should also update the
Vacancy.';
    SET "AWRstat" = 1
  END IF;
  IF "AWRstat" IN( 1,2 ) THEN
    SELECT
"getquestanswer"("string('V',"vacancy"."departmentid"),'AWR',"vacancy"."vac
ancyid"),"LECode","company"."companyid" INTO "LinkingQuestion","LE","coid"
FROM "placement" KEY JOIN "employment" KEY JOIN "company" KEY JOIN
"AWRCompany","placement" KEY JOIN "vacancy" WHERE "placement"."placementid"
= "placeid";
    IF "LE" IS NOT NULL THEN
      SELECT FIRST "AWRJobMasterID" INTO "AWRLink" FROM "AWRJobMaster" KEY
JOIN "placement" KEY JOIN("vacancy","employment" AS "e")
      -- find "master" linked record in case some already linked
      WHERE "AWRJobMasterID" = "AWRLinkCode"
      -- LE code check
      AND EXISTS(SELECT "CompanyID" FROM "AWRCompany" WHERE "LECode" =
"LE" AND "companyid" = "e"."companyid")
      -- job title or question check
      AND
"getquestanswer"("string('V',"vacancy"."departmentid"),'AWR',"vacancy"."vac
ancyid") = "LinkingQuestion"
      AND "LinkingQuestion" <> ''
      AND "e"."personid" = "persid"
    ELSE
```



```
SELECT FIRST "AWRJobMasterID" INTO "AWRLink" FROM "AWRJobMaster" KEY
JOIN "placement" KEY JOIN("vacancy","employment" AS "e")
  -- find "master" linked record in case some already linked
WHERE "AWRJobMasterID" = "AWRLinkCode"
  -- LE code check
AND "e"."companyid" = "coid"
  -- job title or question check
AND "isnull"("e"."position","vacancy"."position") =
"LinkingQuestion"
  AND "LinkingQuestion" <> ''
  AND "e"."personid" = "persid"
END IF END IF;
CASE "AWRStat"
WHEN 1 THEN
  SET "newid" = "uniquekey"("placeid");
  INSERT INTO "AWRJobmaster"(
"awrjobmasterid","placementid","grade","speciality","vacancyid","personid","
AWRLinkCode" ) VALUES(
"newid","placeid",@grade,@speciality,"vacid","persid","AWRLink" ) ;
  SELECT 'N','Y','Y',"newid","UMess",' '
WHEN 2 THEN
  SET "newid" = "uniquekey"("placeid");
  INSERT INTO "AWRJobmaster"(
"awrjobmasterid","placementid","grade","speciality","vacancyid","personid","
AWRLinkCode" ) VALUES(
"newid","placeid",@grade,@speciality,"vacid","persid","AWRLink" ) ;
  SELECT 'N','Y','N',"newid",'',' '
WHEN 3 THEN
  SELECT 'Y','Y','Y',' ',' ',' '
END CASE
END IF;
IF "BookOrConfirmorPlaceorTSflag" = 'T' THEN
  SELECT 'Y','Y','Y',' ',' ',' '
END IF; -- testing code   select 'N','Y','Y',(select first awrjobmasterid
from awrjobmaster where vacancyid is not null and personid is not null),'You
must ask about other agency work and must check the AWR details temp
confirm',' '
IF "BookOrConfirmorPlaceorTSflag" = 'S' THEN
  SELECT 'Y','Y','Y',' ',' ',' '
END IF; -- testing code   select 'N','N','Y',(select first awrjobmasterid
from awrjobmaster where vacancyid is not null and personid is not null),'You
must ask about other agency work and must check the AWR details shift
timesheet',' '
IF "BookOrConfirmorPlaceorTSflag" = 'L' THEN
  IF "placeid" IS NULL THEN
    SELECT 'N','N','N',NULL,"string"('For AWR compatability if this
timesheet relates to a vacancy you must create a placement before entering a
timesheet.','"char"(13),"char"(10),' Go back, create the placement and try
```



```
again. '), '' ;
    RETURN
END IF;
SELECT "employment"."personid", "placement"."vacancyid" INTO
"persid", "vacid" FROM "placement" KEY JOIN "employment" WHERE
"placement"."placementid" = "placeid";
IF EXISTS(SELECT * FROM "person" WHERE "personid" = "persid" AND
"isnull"("directlyemployed",0) >= 1) THEN
    SELECT 'Y', 'Y', 'Y', '', '', '';
    RETURN
END IF;
SELECT FIRST "AWRVacancy"."AWRStatus" INTO "AWRstat" FROM "AWRVacancy"
JOIN "placement" ON "AWRVacancy"."vacancyid" = "placement"."vacancyid" WHERE
"placement"."placementid" = "placeid";
SELECT FIRST "AWRStatus", "AWRjobmasterid" INTO "JAWRstat", "jobmid" FROM
"AWRjobmaster" WHERE "placementid" = "placeid";
-- speciality and grade
IF "JAWRstat" IS NULL THEN
    SET "JAWRstat" = 4
ELSE
    IF "AWRWeeks"("jobmid", "WeekNo", NULL, NULL, NULL, NULL) =
"WeeksToWarnTimesheet" THEN
        SET "Umess" = "string"('This temp will qualify in ', 12 -
"WeeksToWarnTimesheet", ' week(s). Check the AWR Details for what needs to be
done. Check for placements that should be linked or for work for other
Agencies.')
    END IF
END IF;
-- 1 not known 2 applies 3 not
CASE "JAWRstat"
WHEN 1 THEN
    -- need to check work so far here
    IF "AWRGap"("jobmid", "WeekNo", NULL, NULL, NULL, NULL) >= 6 THEN
        SELECT 'N', 'Y', 'Y', "jobmid", '', ''
    ELSE
        SELECT 'N', 'N', 'Y', "jobmid", "UMess", ''
    END IF
WHEN 2 THEN
    IF "AWRGap"("jobmid", "WeekNo", NULL, NULL, NULL, NULL) >= 6 THEN
        SELECT 'N', 'Y', 'N', "jobmid", '', ''
    ELSE
        SELECT 'N', 'N', 'N', "jobmid", "UMess", ''
    END IF
WHEN 3 THEN
    SELECT 'Y', 'Y', 'Y', '', '', ''
WHEN 4 THEN
    -- no record check vac might have changed
    IF "AWRstat" = 3 THEN
        SELECT 'Y', 'Y', 'Y', '', '', ''
    ELSE
        SET "newid" = "uniquekey"("placeid");
```



```
SELECT
"getquestanswer"("string"('V',"vacancy"."departmentid"),'AWR',"vacancy"."vac
ancyid"),"LECode","company"."companyid" INTO "LinkingQuestion","LE","coid"
FROM "placement" KEY JOIN "employment" KEY JOIN "company" KEY JOIN
"AWRCompany","placement" KEY JOIN "vacancy" WHERE "placement"."placementid"
= "placeid";
IF "LE" IS NOT NULL THEN
SELECT FIRST "AWRJobMasterID" INTO "AWRLink" FROM "AWRJobMaster"
KEY JOIN "placement" KEY JOIN("vacancy","employment" AS "e")
-- find "master" linked record in case some already linked
WHERE "AWRJobMasterID" = "AWRLinkCode"
-- LE code check
AND EXISTS(SELECT "CompanyID" FROM "AWRCompany" WHERE "LECode" =
"LE" AND "companyid" = "e"."companyid")
-- job title or question check
AND "isnull"("e"."position","vacancy"."position") =
"LinkingQuestion"
AND "LinkingQuestion" <> ''
AND "e"."personid" = "persid"
ELSE
SELECT FIRST "AWRJobMasterID" INTO "AWRLink" FROM "AWRJobMaster"
KEY JOIN "placement" KEY JOIN("vacancy","employment" AS "e")
-- find "master" linked record in case some already linked
WHERE "AWRJobMasterID" = "AWRLinkCode"
-- LE code check
AND "e"."companyid" = "coid"
-- job title or question check
AND
"getquestanswer"("string"('V',"vacancy"."departmentid"),'AWR',"vacancy"."vac
ancyid") = "LinkingQuestion"
AND "LinkingQuestion" <> ''
AND "e"."personid" = "persid"
END IF;
INSERT INTO "AWRJobmaster"(
"awrjobmasterid","placementid","grade","speciality","vacancyid","personid","
AWRLinkCode" ) VALUES(
"newid","placeid",@grade,@speciality,"vacid","persid","AWRLink" ) ;
SELECT 'N','Y','Y',"newid",'',''
END IF
END CASE
END IF
END
}
```



From:

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

Permanent link:

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

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

