



pears.Pay_Employment



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

Description

Contains records of periods of employment of temps by the Agency (operating the Pears system) itself.

Columns

Column	Type	Null	Default	Comment
Pay_EmploymentID	char(20)	NOT NULL		
PersonID	char(20)	NOT NULL		
StartDate	date	NULL		
EndDate	date	NULL		
DateContractReturned	date	NULL		
ExpenseBenefitOptOutID	char(20)	NULL		
MultiSiteWorkingAgreement	smallint	NULL		
payrollnumber	char(15)	NULL		
TaxMethod	tinyint	NULL		1=PAYE 2=Ltd 3=Selfemp
CompanyName	char(50)	NULL		
CompanyRegNo	char(20)	NULL		
HMRCEngagementDetail	char(1)	NULL		
UTR	char(20)	NULL		
PayrollAddr1	char(40)	NULL		
PayrollAddr2	char(40)	NULL		
PayrollAddr3	char(40)	NULL		
PayrollAddr4	char(40)	NULL		
PayrollPostcode	char(30)	NULL		
PayrollFullAddr	long varchar	NULL		
VATRegistered	tinyint	NULL		
VATNumber	char(50)	NULL		
CompositeID	char(20)	NULL		



Primary Key

- Pay_EmploymentID

Foreign Keys

Constraint	Columns	References	Delete/update action
person	PersonID	pears.Person (personid)	NOT NULL;
ExpenseBenefitOptOut	ExpenseBenefitOptOutID	pears.ExpenseBenefitOptOut (ExpenseBenefitOptOutID)	

Referenced By

Table	Constraint	Columns	Referenced columns
pears.TempTimeSheet	Pay_Employment	Pay_EmploymentID	Pay_EmploymentID

Indexes

Name	Type	Columns	Detail
WeEmploy_PersonID	Index	PersonID	
payemp_startdate	Index	StartDate	
pay_employment_taxmethod_enddate	Index	TaxMethod, EndDate	
pay_employment_payrollnumber	Index	payrollnumber	

Triggers

Name	Timing	Event
PayEmploymentAudit	after	update of "EndDate", "StartDate", "CompanyName", "CompanyRegNo", "HMRCEngagementDetail", "UTR", "VATNumber", "VATRegistered" order 1
PayEmploymentAuditInsert	after	insert order 1
PayEmploymentInsert	before	insert order 2

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."Pay_Employment"
-- Table comment: Contains records of periods of employment of temps by the
Agency (operating the Pears system) itself.
-- Statement count: 15
```

```
CREATE TABLE "pears"."Pay_Employment" (
```



```
"Pay_EmploymentID"      CHAR(20) NOT NULL
,"PersonID"             CHAR(20) NOT NULL
,"StartDate"            DATE NULL
,"EndDate"              DATE NULL
,"DateContractReturned" DATE NULL
,"ExpenseBenefitOptOutID" CHAR(20) NULL
,"MultiSiteWorkingAgreement" SMALLINT NULL
,"payrollnumber"        CHAR(15) NULL
,"TaxMethod"            tinyint NULL
,"CompanyName"          CHAR(50) NULL
,"CompanyRegNo"         CHAR(20) NULL
,"HMRCEngagementDetail" CHAR(1) NULL
,"UTR"                  CHAR(20) NULL
,"PayrollAddr1"         CHAR(40) NULL
,"PayrollAddr2"         CHAR(40) NULL
,"PayrollAddr3"         CHAR(40) NULL
,"PayrollAddr4"         CHAR(40) NULL
,"PayrollPostcode"      CHAR(30) NULL
,"PayrollFullAddr"      long VARCHAR NULL
,"VATRegistered"        tinyint NULL
,"VATNumber"            CHAR(50) NULL
,"CompositeID"          CHAR(20) NULL
,PRIMARY KEY ("Pay_EmploymentID" ASC)
)
GO

COMMENT ON COLUMN "pears"."Pay_Employment"."TaxMethod" IS
'1=PAYE 2=Ltd 3=Selfemp'
GO

COMMENT ON TABLE "pears"."Pay_Employment" IS
'Contains records of periods of employment of temps by the Agency
(operating the Pears system) itself.'
GO

ALTER TABLE "pears"."Pay_Employment"
ADD NOT NULL FOREIGN KEY "person" ("PersonID" ASC)
REFERENCES "pears"."Person" ("personid")
GO

ALTER TABLE "pears"."Pay_Employment"
ADD FOREIGN KEY "ExpenseBenefitOptOut" ("ExpenseBenefitOptOutID" ASC)
REFERENCES "pears"."ExpenseBenefitOptOut" ("ExpenseBenefitOptOutID")
GO
```



```
CREATE INDEX "WeEmploy_PersonID" ON "pears"."Pay_Employment"
    ( "PersonID" )
GO

CREATE INDEX "payemp_startdate" ON "pears"."Pay_Employment"
    ( "StartDate" )
GO

CREATE INDEX "pay_employment_taxmethod_enddate" ON "pears"."Pay_Employment"
    ( "TaxMethod", "EndDate" )
GO

CREATE INDEX "pay_employment_payrollnumber" ON "pears"."Pay_Employment"
    ( "payrollnumber" )
GO

CREATE TRIGGER "PayEmploymentAudit" after UPDATE OF "EndDate",
"StartDate", "CompanyName", "CompanyRegNo", "HMRCEngagementDetail", "UTR", "VATNu
mber",
"VATRegistered" ORDER 1 ON "pears"."Pay_Employment"
REFERENCING OLD AS "old_name" NEW AS "new_name"
FOR each ROW
BEGIN
    DECLARE "PName" CHAR(250);
    SET "PName" = (SELECT "Name" FROM "Person" WHERE "PersonID" =
"isnull"("old_name"."PersonID", "new_name"."PersonID"));
    IF UPDATE("StartDate") AND (EXISTS (SELECT * FROM "AuditItems" WHERE
"AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName" = 'Employment Start
Date')) THEN
        CALL "AuditLog"('PERSON', "old_name"."personid", "string"('Employment
Start Date Updated -
', "PName"), "string"("old_name"."startdate"), "string"("new_name"."startdate")
)
    END IF;
    IF UPDATE("EndDate") AND (EXISTS (SELECT * FROM "AuditItems" WHERE
"AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName" = 'Employment End
Date')) THEN
        CALL "AuditLog"('PERSON', "old_name"."personid", "string"('Employment End
Date Updated -
', "PName"), "string"("old_name"."enddate"), "string"("new_name"."enddate"))
    END IF;
    IF UPDATE("CompanyName") AND (EXISTS (SELECT * FROM "AuditItems" WHERE
```



```
"AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName" = 'Employment
Company Name')) THEN
    CALL "AuditLog"('PERSON',"old_name"."personid","string"('Employment
Company Name -
',"PName"),"string"("old_name"."CompanyName"),"string"("new_name"."CompanyNa
me"))
    END IF;
    IF UPDATE("CompanyRegNo") AND(EXISTS(SELECT * FROM "AuditItems" WHERE
"AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName" = 'Employment
Company Reg. No')) THEN
        CALL "AuditLog"('PERSON',"old_name"."personid","string"('Employment
Company Reg. No -
',"PName"),"string"("old_name"."CompanyRegNo"),"string"("new_name"."CompanyR
egNo"))
        END IF;
        IF UPDATE("HMRCEngagementDetail") AND(EXISTS(SELECT * FROM "AuditItems"
WHERE "AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName" = 'Employment
HMRC Engagement Details')) THEN
            CALL "AuditLog"('PERSON',"old_name"."personid","string"('Employment HMRC
Engagement Details -
',"PName"),"string"("old_name"."HMRCEngagementDetail"),"string"("new_name"."
HMRCEngagementDetail"))
            END IF;
            IF UPDATE("UTR") AND(EXISTS(SELECT * FROM "AuditItems" WHERE "AreaName" =
'Person' AND "AuditFlag" = 1 AND "ItemName" = 'Employment Unique Tax
Reference')) THEN
                CALL "AuditLog"('PERSON',"old_name"."personid","string"('Employment
Unique Tax Reference -
',"PName"),"string"("old_name"."UTR"),"string"("new_name"."UTR"))
                END IF;
                IF UPDATE("VATNumber") AND(EXISTS(SELECT * FROM "AuditItems" WHERE
"AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName" = 'Employment Vat
Number')) THEN
                    CALL "AuditLog"('PERSON',"old_name"."personid","string"('Employment Vat
Number -
',"PName"),"string"("old_name"."VATNumber"),"string"("new_name"."VATNumber")
)
                    END IF;
                    IF UPDATE("VATRegistered") AND(EXISTS(SELECT * FROM "AuditItems" WHERE
"AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName" = 'Employment Vat
Registered')) THEN
                        CALL "AuditLog"('PERSON',"old_name"."personid","string"('Employment Vat
Registered -
',"PName"),"string"("old_name"."VATRegistered"),"string"("new_name"."VATRegi
stered"))
                        END IF
                    END
                GO
```



```
COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."Pay_Employment"."PayEmploymentAudit" IS
{CREATE TRIGGER PayEmploymentAudit
  after UPDATE OF EndDate, StartDate,
  CompanyName, CompanyRegNo, HMRCEngagementDetail, UTR, VATNumber, VATRegistered
  ORDER 1 ON pears.Pay_Employment
  REFERENCING OLD AS old_name NEW AS new_name
  FOR each ROW
  BEGIN
    DECLARE PName CHAR(250);
    SET PName=(SELECT Name FROM Person WHERE PersonID =
  isnull(old_name.PersonID,new_name.PersonID));
    IF UPDATE (StartDate) AND (EXISTS(SELECT * FROM AuditItems WHERE AreaName
  = 'Person' AND AuditFlag = 1 AND ItemName ='Employment Start Date')) THEN
      CALL AuditLog('PERSON',old_name.personid,string('Employment Start Date
  Updated - ',PName),string(old_name.startdate),string(new_name.startdate))
    END IF;
    IF UPDATE (EndDate) AND (EXISTS(SELECT * FROM AuditItems WHERE AreaName =
  'Person' AND AuditFlag = 1 AND ItemName ='Employment End Date')) THEN
      CALL AuditLog('PERSON',old_name.personid,string('Employment End Date
  Updated - ',PName),string(old_name.enddate),string(new_name.enddate))
    END IF;
    IF UPDATE (CompanyName) AND (EXISTS(SELECT * FROM AuditItems WHERE
  AreaName = 'Person' AND AuditFlag = 1 AND ItemName ='Employment Company
  Name')) THEN
      CALL AuditLog('PERSON',old_name.personid,string('Employment Company Name
  - ',PName),string(old_name.CompanyName),string(new_name.CompanyName))
    END IF;
    IF UPDATE (CompanyRegNo) AND (EXISTS(SELECT * FROM AuditItems WHERE
  AreaName = 'Person' AND AuditFlag = 1 AND ItemName ='Employment Company Reg.
  No')) THEN
      CALL AuditLog('PERSON',old_name.personid,string('Employment Company Reg.
  No - ',PName),string(old_name.CompanyRegNo),string(new_name.CompanyRegNo))
    END IF;
    IF UPDATE (HMRCEngagementDetail) AND (EXISTS(SELECT * FROM AuditItems
  WHERE AreaName = 'Person' AND AuditFlag = 1 AND ItemName ='Employment HMRC
  Engagement Details')) THEN
      CALL AuditLog('PERSON',old_name.personid,string('Employment HMRC
  Engagement Details -
  ',PName),string(old_name.HMRCEngagementDetail),string(new_name.HMRCEngagemen
  tDetail))
    END IF;
    IF UPDATE (UTR) AND (EXISTS(SELECT * FROM AuditItems WHERE AreaName =
  'Person' AND AuditFlag = 1 AND ItemName ='Employment Unique Tax Reference'))
  THEN
      CALL AuditLog('PERSON',old_name.personid,string('Employment Unique Tax
```



```
Reference - ',PName),string(old_name.UTR),string(new_name.UTR))
END IF;
IF UPDATE (VATNumber) AND (EXISTS(SELECT * FROM AuditItems WHERE AreaName
= 'Person' AND AuditFlag = 1 AND ItemName ='Employment Vat Number')) THEN
CALL AuditLog('PERSON',old_name.personid,string('Employment Vat Number -
',PName),string(old_name.VATNumber),string(new_name.VATNumber))
END IF;
IF UPDATE (VATRegistered) AND (EXISTS(SELECT * FROM AuditItems WHERE
AreaName = 'Person' AND AuditFlag = 1 AND ItemName ='Employment Vat
Registered')) THEN
CALL AuditLog('PERSON',old_name.personid,string('Employment Vat
Registered -
',PName),string(old_name.VATRegistered),string(new_name.VATRegistered))
END IF;
END
}
GO

CREATE TRIGGER "PayEmploymentAuditInsert" after INSERT ORDER 1 ON
"pears"."Pay_Employment"
REFERENCING NEW AS "new_name"
FOR each ROW
BEGIN
DECLARE "PName" CHAR(250);
SET "PName" = (SELECT "Name" FROM "Person" WHERE "PersonID" =
"new_name"."PersonID");
IF "new_name"."StartDate" IS NOT NULL AND(EXISTS(SELECT * FROM
"AuditItems" WHERE "AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName"
= 'Employment Start Date')) THEN
CALL "AuditLog"('PERSON',"new_name"."personid",string('Employment
Start Date - ', "PName"),'',string("new_name"."startdate"))
END IF;
IF "new_name"."EndDate" IS NOT NULL AND(EXISTS(SELECT * FROM "AuditItems"
WHERE "AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName" = 'Employment
End Date')) THEN
CALL "AuditLog"('PERSON',"new_name"."personid",string('Employment End
Date - ', "PName"),'',string("new_name"."enddate"))
END IF;
IF "new_name"."CompanyName" IS NOT NULL AND(EXISTS(SELECT * FROM
"AuditItems" WHERE "AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName"
= 'Employment Company Name')) THEN
CALL "AuditLog"('PERSON',"new_name"."personid",string('Employment
Company Name - ', "PName"),'',string("new_name"."CompanyName"))
END IF;
IF "new_name"."CompanyRegNo" IS NOT NULL AND(EXISTS(SELECT * FROM
"AuditItems" WHERE "AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName"
= 'Employment Company Reg. No')) THEN
```



```
CALL "AuditLog"('PERSON',"new_name"."personid","string"('Employment
Company Reg. No - ', "PName"),'', "string"("new_name"."CompanyRegNo"))
END IF;
IF "new_name"."HMRCEngagementDetail" IS NOT NULL AND(EXISTS(SELECT * FROM
"AuditItems" WHERE "AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName"
= 'Employment HMRC Engagement Details')) THEN
CALL "AuditLog"('PERSON',"new_name"."personid","string"('Employment HMRC
Engagement Details -
', "PName"),'', "string"("new_name"."HMRCEngagementDetail"))
END IF;
IF "new_name"."UTR" IS NOT NULL AND(EXISTS(SELECT * FROM "AuditItems"
WHERE "AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName" = 'Employment
Unique Tax Reference')) THEN
CALL "AuditLog"('PERSON',"new_name"."personid","string"('Employment
Unique Tax Reference - ', "PName"),'', "string"("new_name"."UTR"))
END IF;
IF "new_name"."VATNumber" IS NOT NULL AND(EXISTS(SELECT * FROM
"AuditItems" WHERE "AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName"
= 'Employment Vat Number')) THEN
CALL "AuditLog"('PERSON',"new_name"."personid","string"('Employment Vat
Number - ', "PName"),'', "string"("new_name"."VATNumber"))
END IF;
IF "new_name"."VATRegistered" IS NOT NULL AND(EXISTS(SELECT * FROM
"AuditItems" WHERE "AreaName" = 'Person' AND "AuditFlag" = 1 AND "ItemName"
= 'Employment Vat Registered')) THEN
CALL "AuditLog"('PERSON',"new_name"."personid","string"('Employment Vat
Registered - ', "PName"),'', "string"("new_name"."VATRegistered"))
END IF
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."Pay_Employment"."PayEmploymentAuditInsert" IS
{CREATE TRIGGER PayEmploymentAuditInsert
AFTER INSERT
ORDER 1 ON pears.Pay_Employment
REFERENCING NEW AS new_name
FOR each ROW
BEGIN
DECLARE PName CHAR(250);
SET PName=(SELECT Name FROM Person WHERE PersonID = new_name.PersonID);
IF new_name.StartDate IS NOT NULL AND (EXISTS(SELECT * FROM AuditItems
WHERE AreaName = 'Person' AND AuditFlag = 1 AND ItemName = 'Employment Start
Date')) THEN
CALL AuditLog('PERSON',new_name.personid,string('Employment Start Date -
', PName),'', string(new_name.startdate))
END IF;
```




```
IF new_name.EndDate IS NOT NULL AND (EXISTS(SELECT * FROM AuditItems WHERE
AreaName = 'Person' AND AuditFlag = 1 AND ItemName ='Employment End Date'))
THEN
    CALL AuditLog('PERSON',new_name.personid,string('Employment End Date -
',PName),'',string(new_name.enddate))
    END IF;
IF new_name.CompanyName IS NOT NULL AND (EXISTS(SELECT * FROM AuditItems
WHERE AreaName = 'Person' AND AuditFlag = 1 AND ItemName ='Employment
Company Name')) THEN
    CALL AuditLog('PERSON',new_name.personid,string('Employment Company Name
- ',PName),'',string(new_name.CompanyName))
    END IF;
IF new_name.CompanyRegNo IS NOT NULL AND (EXISTS(SELECT * FROM AuditItems
WHERE AreaName = 'Person' AND AuditFlag = 1 AND ItemName ='Employment
Company Reg. No')) THEN
    CALL AuditLog('PERSON',new_name.personid,string('Employment Company Reg.
No - ',PName),'',string(new_name.CompanyRegNo))
    END IF;
IF new_name.HMRCEngagementDetail IS NOT NULL AND (EXISTS(SELECT * FROM
AuditItems WHERE AreaName = 'Person' AND AuditFlag = 1 AND ItemName
='Employment HMRC Engagement Details')) THEN
    CALL AuditLog('PERSON',new_name.personid,string('Employment HMRC
Engagement Details - ',PName),'',string(new_name.HMRCEngagementDetail))
    END IF;
IF new_name.UTR IS NOT NULL AND (EXISTS(SELECT * FROM AuditItems WHERE
AreaName = 'Person' AND AuditFlag = 1 AND ItemName ='Employment Unique Tax
Reference')) THEN
    CALL AuditLog('PERSON',new_name.personid,string('Employment Unique Tax
Reference - ',PName),'',string(new_name.UTR))
    END IF;
IF new_name.VATNumber IS NOT NULL AND (EXISTS(SELECT * FROM AuditItems
WHERE AreaName = 'Person' AND AuditFlag = 1 AND ItemName ='Employment Vat
Number')) THEN
    CALL AuditLog('PERSON',new_name.personid,string('Employment Vat Number -
',PName),'',string(new_name.VATNumber))
    END IF;
IF new_name.VATRegistered IS NOT NULL AND (EXISTS(SELECT * FROM AuditItems
WHERE AreaName = 'Person' AND AuditFlag = 1 AND ItemName ='Employment Vat
Registered')) THEN
    CALL AuditLog('PERSON',new_name.personid,string('Employment Vat
Registered - ',PName),'',string(new_name.VATRegistered))
    END IF;
END
}
GO
```

```
CREATE TRIGGER "PayEmploymentInsert" BEFORE INSERT ORDER 2 ON
```



```
"pears"."Pay_Employment"
REFERENCING NEW AS "new_name"
FOR each ROW
BEGIN
    DECLARE "PName" CHAR(250);
    SET "new_name"."compositeid" = (SELECT "compositeid" FROM "pay_employee"
AS "ee" WHERE "ee"."personid" = "new_name"."personid")
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."Pay_Employment"."PayEmploymentInsert" IS
{CREATE TRIGGER PayEmploymentInsert
    BEFORE INSERT
ORDER 2 ON pears.Pay_Employment
REFERENCING NEW AS new_name
FOR each ROW
BEGIN
    DECLARE PName CHAR(250);
    SET new_name.compositeid = (SELECT compositeid FROM pay_employee ee WHERE
ee.personid = new_name.personid)
END
}
GO
```

From:

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

Permanent link:

https://iqxusers.co.uk/iqxhelp/doku.php?id=database:tables:pears_pay_employment

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

