pears.Pay_Employee

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

Description

Data related to a persons employment by the agency. One to one relationship with Person. Only created when required.

Columns

Column	Type	Null	Default	Comment
PersonID	char(20)	NOT NULL		
P45Requested	smallint	NULL	0	
TempaidRecordRequired	smallint	NULL	1	
BankName	char(30)	NULL		
BankAddr1	char(30)	NULL		
BankAddr2	char(30)	NULL		
BankSortCode	char(8)	NULL		
BankAcName	char(50)	NULL		
BankAcNo	char(15)	NULL		
BankRef	char(20)	NULL		
HolidayPayNotes	long varchar	NULL		
PayMethod	tinyint	NULL	1	1=Bacs 2=Cash 3=Cheque 4=Chaps 5=Intl
PayFrequency	tinyint	NULL	52	52=weekly 4=Qtrly 12=Monthly 13=4Wkly 25=2Wkly
TaxMethod	tinyint	NULL	1	1=PAYE 2=Ltd 3=Selfemp
CompanyName	char(50)	NULL		
RegNo	char(20)	NULL		
VATRegistered	tinyint	NULL	0	
VATNumber	char(20)	NULL		
TimesheetsWithPayslip	tinyint	NULL	0	
TransferBatch	integer	NULL		
ExtNumber	integer	NULL		

Column	Type	Null	Default	Comment
TaxCode	char(6)	NULL		
NILetter	char(1)	NULL		
Week1Ind	smallint	NULL		
ExtraCode	char(30)	NULL		
SecondaryAgencyID	char(20)	NULL		If candidate is registered with a Secondary Agency company
SecondaryAgencyRef	char(20)	NULL		
CISTemplateID	char(20)	NULL		
CompositeID	char(20)	NULL		
CompositePayCoReference	char(60)	NULL		
IBAN	char(34)	NULL		
SWIFTBIC	char(11)	NULL		
UniqueTaxReference	char(20)	NULL		
CISVerificationNumber	char(20)	NULL		
CISTaxTreatment	char(10)	NULL		
CISVerificationDate	date	NULL		
CIS07TemplateID	char(20)	NULL		
CISEffectiveDate	date	NULL		
CurrencyCode	char(3)	NULL		Used for exporting to banking systems
EmailPayslips	tinyint	NULL	0	
TextPaymentAdvice	tinyint	NULL	0	
NoHolidayPay	smallint	NULL		
WhenEntered	timestamp	NULL	current timestamp	
WhoEntered	char(20)	NULL		
ExcludeForceNI	smallint	NULL	0	Visible if ForceNI is on. Use to exclude from NI invoicing when LTD tax method
HMRCEngagementDetail	char(1)	NULL	'Z'	
PayrollAddr1	char(40)	NULL		
PayrollAddr2	char(40)	NULL		
PayrollAddr3	char(40)	NULL		
PayrollAddr4	char(40)	NULL		
PayrollPostcode	char(30)	NULL		
ForeignAwaitingNI	tinyint	NULL	0	
SupplierCode	char(12)	NULL		Self bill account for umbrella etc.
SecondaryAgencyCandidateStatus	char(1)	NULL		
LSM_TransferBatch	integer	NULL		

Primary Key

- PersonID

Foreign Keys

Constraint	Columns	References	Delete/update action
person	PersonID	pears.Person (personid)	NOT NULL;
Company	SecondaryAgencyID	pears.Company (companyid)	ON DELETE SET NULL
CISTemplate	CISTemplateID	pears.CISTemplate (CISTemplateID)	ON DELETE SET NULL
Composite	CompositeID	pears.Composite (CompositeId)	ON DELETE SET NULL
CIS07Template	CIS07TemplateID	pears.CIS07Template (CIS07TemplateID)	ON DELETE CASCADE
SecondaryAgencyCandidateStatus	SecondaryAgencyCandidateStatus	pears.SecondaryAgencyCandidateStatus (SecondaryAgencyCandidateStatusID)	

Referenced By

- No incoming foreign keys found.

Indexes

Name	Type	Columns	Detail
payemp_extnumber	Index	ExtNumber	
payemp_transferbatch	Index	TransferBatch	
payemp_p45requested	Index	P45Requested	

Triggers

Name	Timing	Event
payemployeeupdate	before	update order 1
payemployeeinsert	before	insert order 2
PayEmployeeAudit	before	update of “PayMethod”, “CompositeID”,“TaxCode”,“TaxMethod”,“NILetter”,“VATNumber”,“VATRegistered”,“CompanyName”,“RegNo”,“ExtNumber”,“BankSortCode”,“BankAcNo”,“Suppliercode”, “BankAcName”,“BankRef”,“UniqueTaxReference”,“CISTaxTreatment”,“CISVerificationDate”,“CISVerificationNumber”,“CISEffectiveDate”,“HMRCEngagementDetail” order 3

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."Pay_Employee"
-- Table comment: Data related to a persons employment by the agency. One to one relationship with Person. Only created when required.
```

```
-- Statement count: 24

CREATE TABLE "pears"."Pay_Employee" (
    "PersonID"                      CHAR(20) NOT NULL
   ,"P45Requested"                  SMALLINT NULL DEFAULT 0
   ,"TempaidRecordRequired"         SMALLINT NULL DEFAULT 1
   ,"BankName"                      CHAR(30) NULL
   ,"BankAddr1"                     CHAR(30) NULL
   ,"BankAddr2"                     CHAR(30) NULL
   ,"BankSortCode"                  CHAR(8) NULL
   ,"BankAcName"                    CHAR(50) NULL
   ,"BankAcNo"                      CHAR(15) NULL
   ,"BankRef"                       CHAR(20) NULL
   ,"HolidayPayNotes"               long VARCHAR NULL
   ,"PayMethod"                     tinyint NULL DEFAULT 1
   ,"PayFrequency"                  tinyint NULL DEFAULT 52
   ,"TaxMethod"                     tinyint NULL DEFAULT 1
   ,"CompanyName"                   CHAR(50) NULL
   ,"RegNo"                         CHAR(20) NULL
   ,"VATRegistered"                 tinyint NULL DEFAULT 0
   ,"VATNumber"                     CHAR(20) NULL
   ,"TimesheetsWithPayslip"         tinyint NULL DEFAULT 0
   ,"TransferBatch"                 INTEGER NULL
   ,"ExtNumber"                     INTEGER NULL
   ,"TaxCode"                       CHAR(6) NULL
   ,"NILetter"                      CHAR(1) NULL
   ,"Week1Ind"                      SMALLINT NULL
   ,"ExtraCode"                     CHAR(30) NULL
   ,"SecondaryAgencyID"             CHAR(20) NULL
   ,"SecondaryAgencyRef"            CHAR(20) NULL
   ,"CISTemplateID"                 CHAR(20) NULL
   ,"CompositeID"                   CHAR(20) NULL
   ,"CompositePayCoReference"       CHAR(60) NULL
   ,"IBAN"                          CHAR(34) NULL
   ,"SWIFTBIC"                      CHAR(11) NULL
   ,"UniqueTaxReference"            CHAR(20) NULL
   ,"CISVerificationNumber"         CHAR(20) NULL
   ,"CISTaxTreatment"               CHAR(10) NULL
   ,"CISVerificationDate"           DATE NULL
   ,"CIS07TemplateID"               CHAR(20) NULL
   ,"CISEffectiveDate"              DATE NULL
   ,"CurrencyCode"                  CHAR(3) NULL
   ,"EmailPayslips"                 tinyint NULL DEFAULT 0
   ,"TextPaymentAdvice"             tinyint NULL DEFAULT 0
   ,"NoHolidayPay"                  SMALLINT NULL
   ,"WhenEntered"                   TIMESTAMP NULL DEFAULT CURRENT
TIMESTAMP
   ,"WhoEntered"                    CHAR(20) NULL
```

```
    ,"ExcludeForceNI"                   SMALLINT NULL DEFAULT 0
    ,"HMRCEngagementDetail"             CHAR(1) NULL DEFAULT 'Z'
    ,"PayrollAddr1"                     CHAR(40) NULL
    ,"PayrollAddr2"                     CHAR(40) NULL
    ,"PayrollAddr3"                     CHAR(40) NULL
    ,"PayrollAddr4"                     CHAR(40) NULL
    ,"PayrollPostcode"                  CHAR(30) NULL
    ,"ForeignAwaitingNI"                tinyint NULL DEFAULT 0
    ,"SupplierCode"                     CHAR(12) NULL
    ,"SecondaryAgencyCandidateStatus" CHAR(1) NULL
    ,"LSM_TransferBatch"                INTEGER NULL
    ,PRIMARY KEY ("PersonID" ASC)
)
GO


COMMENT ON COLUMN "pears"."Pay_Employee"."PayMethod" IS
    '1=Bacs 2=Cash 3=Cheque 4=Chaps 5=Intl'
GO


COMMENT ON COLUMN "pears"."Pay_Employee"."PayFrequency" IS
    '52=weekly 4=Qtrly 12=Monthly 13=4Wkly 25=2Wkly'
GO


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


COMMENT ON COLUMN "pears"."Pay_Employee"."SecondaryAgencyID" IS
    'If candidate is registered with a Secondary Agency company'
GO


COMMENT ON COLUMN "pears"."Pay_Employee"."CurrencyCode" IS
    'Used for exporting to banking systems'
GO


COMMENT ON COLUMN "pears"."Pay_Employee"."ExcludeForceNI" IS
    'Visible if ForceNI is on. Use to exclude from NI invoicing when LTD tax
method'
GO


COMMENT ON COLUMN "pears"."Pay_Employee"."SupplierCode" IS
```

```
    'Self bill account for umbrella etc.'
GO


COMMENT ON TABLE "pears"."Pay_Employee" IS
    'Data related to a persons employment by the agency. One to one
relationship with Person. Only created when required.'
GO


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


ALTER TABLE "pears"."Pay_Employee"
    ADD FOREIGN KEY "Company" ("SecondaryAgencyID" ASC)
    REFERENCES "pears"."Company" ("companyid")
    ON DELETE SET NULL
GO


ALTER TABLE "pears"."Pay_Employee"
    ADD FOREIGN KEY "CISTemplate" ("CISTemplateID" ASC)
    REFERENCES "pears"."CISTemplate" ("CISTemplateID")
    ON DELETE SET NULL
GO


ALTER TABLE "pears"."Pay_Employee"
    ADD FOREIGN KEY "Composite" ("CompositeID" ASC)
    REFERENCES "pears"."Composite" ("CompositeId")
    ON DELETE SET NULL
GO


ALTER TABLE "pears"."Pay_Employee"
    ADD FOREIGN KEY "CIS07Template" ("CIS07TemplateID" ASC)
    REFERENCES "pears"."CIS07Template" ("CIS07TemplateID")
    ON DELETE CASCADE
GO


ALTER TABLE "pears"."Pay_Employee"
    ADD FOREIGN KEY "SecondaryAgencyCandidateStatus"
("SecondaryAgencyCandidateStatus" ASC)
    REFERENCES "pears"."SecondaryAgencyCandidateStatus"
```

```
("SecondaryAgencyCandidateStatusID")
GO


CREATE INDEX "payemp_extnumber" ON "pears"."Pay_Employee"
    ( "ExtNumber" )
GO


CREATE INDEX "payemp_transferbatch" ON "pears"."Pay_Employee"
    ( "TransferBatch" )
GO


CREATE INDEX "payemp_p45requested" ON "pears"."Pay_Employee"
    ( "P45Requested" )
GO


CREATE TRIGGER "payemployeeupdate" BEFORE UPDATE ORDER 1 ON
"pears"."Pay_Employee"
REFERENCING OLD AS "old_payemp" NEW AS "new_payemp"
FOR each ROW
BEGIN
  IF "old_payemp"."transferbatch" > 0 THEN
    SET "new_payemp"."transferbatch" = 0;
    UPDATE "accordemployee" SET "transferbatch" = 0 WHERE "personid" =
"old_payemp"."personid"
  END IF;
  IF "new_payemp"."p45requested" = 1 THEN
    UPDATE "person" SET "p46declaration" = 0 WHERE "personid" =
"old_payemp"."personid"
  END IF
END
GO


COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."Pay_Employee"."payemployeeupdate" IS
{CREATE TRIGGER payemployeeupdate
 BEFORE UPDATE ORDER 1 ON
pears.Pay_Employee
REFERENCING OLD AS old_payemp NEW AS new_payemp
FOR each ROW
BEGIN
  IF  old_payemp.transferbatch > 0 THEN
      SET new_payemp.transferbatch = 0;
     UPDATE accordemployee SET transferbatch = 0 WHERE personid =
```

```
old_payemp.personid;
  END IF;
  IF  new_payemp.p45requested = 1 THEN
    UPDATE person SET p46declaration = 0 WHERE personid =
old_payemp.personid
  END IF;
END
}
GO


CREATE TRIGGER "payemployeeinsert" BEFORE INSERT ORDER 2 ON
"pears"."Pay_Employee"
REFERENCING NEW AS "new_payemp"
FOR each ROW
BEGIN
  IF "new_payemp"."taxcode" IS NULL THEN
    SET "new_payemp"."taxcode" = (SELECT FIRST
"isnull"("defaulttaxcode",'BR') FROM "params");
    SET "new_payemp"."week1ind" = (SELECT FIRST
"isnull"("defaultweek1ind",0) FROM "params")
  ELSE
    IF "new_payemp"."week1ind" IS NULL THEN
      SET "new_payemp"."week1ind" = 0
    END IF END IF;
  IF "new_payemp"."niletter" IS NULL THEN
    SET "new_payemp"."niletter" = (SELECT FIRST
"isnull"("defaultniletter",'A') FROM "params")
  END IF;
  SET "new_payemp"."WhoEntered" = "userstaffid"
exception
  WHEN others THEN
    SET "new_payemp"."WhoEntered" = NULL
END
GO


COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."Pay_Employee"."payemployeeinsert" IS
{CREATE TRIGGER payemployeeinsert
 BEFORE INSERT ORDER 2 ON
pears.Pay_Employee
REFERENCING NEW AS new_payemp
FOR each ROW
BEGIN
  IF new_payemp.taxcode IS NULL THEN
    SET new_payemp.taxcode=(SELECT FIRST isnull(defaulttaxcode,'BR') FROM
params);
```

```
    SET new_payemp.week1ind=(SELECT FIRST isnull(defaultweek1ind,0) FROM
params)
  ELSE
    IF new_payemp.week1ind IS NULL THEN
      SET new_payemp.week1ind=0
    END IF
  END IF;
  IF new_payemp.niletter IS NULL THEN
    SET new_payemp.niletter=(SELECT FIRST isnull(defaultniletter,'A') FROM
params)
  END IF;
  SET new_payemp.WhoEntered=userstaffid
exception
  WHEN others THEN
    SET new_payemp.WhoEntered=NULL
END
}
GO


CREATE TRIGGER "PayEmployeeAudit" BEFORE UPDATE OF "PayMethod",
"CompositeID","TaxCode","TaxMethod","NILetter","VATNumber","VATRegistered","
CompanyName","RegNo","ExtNumber","BankSortCode","BankAcNo","Suppliercode",
"BankAcName","BankRef","UniqueTaxReference","CISTaxTreatment","CISVerificati
onDate","CISVerificationNumber","CISEffectiveDate","HMRCEngagementDetail"
ORDER 3 ON
"pears"."Pay_Employee"
REFERENCING OLD AS "old_pers" NEW AS "new_pers"
FOR each ROW
WHEN(EXISTS(SELECT * FROM "AuditItems" WHERE "AreaName" = 'Person Pay' AND
"AuditFlag" = 1))
BEGIN
  DECLARE @AuditList long VARCHAR;
  DECLARE @OldDescrip CHAR(250);
  DECLARE @NewDescrip CHAR(250);
  DECLARE @PersonName CHAR(250);
  SELECT "string"(',',"list"("ItemName"),',') INTO @AuditList FROM
"AuditItems" WHERE "AreaName" = 'Person Pay' AND "AuditFlag" = 1;
  SET @PersonName = (SELECT "Name" FROM "Person" WHERE "PersonID" =
"isnull"("old_pers"."PersonID","new_pers"."PersonID"));
  -- Payment Method
  IF "locate"(@AuditList,',Payment Method,') > 0 AND UPDATE("PayMethod")
THEN
    CALL "AuditLog"('PERSON',"old_pers"."personid","string"('Payment Method
- ',@PersonName),"string"("old_pers"."PayMethod",CASE "old_pers"."PayMethod"
WHEN 1 THEN '- BACS' WHEN 2 THEN '- Cash' WHEN 3 THEN '- Cheque' ELSE '-
Other' END),
    "string"("new_pers"."PayMethod",CASE "new_pers"."PayMethod" WHEN 1 THEN
```

```
'- BACS' WHEN 2 THEN '- Cash' WHEN 3 THEN '- Cheque' ELSE '- Other' END))
  -- Tax Code
  END IF;
  IF "locate"(@AuditList,',Tax Code,') > 0 AND UPDATE("TaxCode") THEN
    CALL "AuditLog"('PERSON',"old_pers"."personid","string"('Tax Code -
',@PersonName),"old_pers"."TaxCode","new_pers"."TaxCode")
  END IF;
  -- Tax Method
  IF "locate"(@AuditList,',Tax Method,') > 0 AND UPDATE("TaxMethod") THEN
    CALL "AuditLog"('PERSON',"old_pers"."personid","string"('Tax Method -
',@PersonName),"string"("old_pers"."TaxMethod",CASE "old_pers"."TaxMethod"
WHEN 1 THEN '- PAYES' WHEN 2 THEN '- Ltd Co' WHEN 3 THEN '- SelfEmp' ELSE '-
Other' END),
    "string"("new_pers"."TaxMethod",CASE "new_pers"."TaxMethod" WHEN 1 THEN
'- PAYES' WHEN 2 THEN '- Ltd Co' WHEN 3 THEN '- SelfEmp' ELSE '- Other'
END))
  -- NI Letter
  END IF;
  IF "locate"(@AuditList,',NI Letter,') > 0 AND UPDATE("NILetter") THEN
    CALL "AuditLog"('PERSON',"old_pers"."personid","string"('NI Letter -
',@PersonName),"old_pers"."NILetter","new_pers"."NILetter")
  END IF;
  -- VAT Number
  IF "locate"(@AuditList,',VAT Number,') > 0 AND UPDATE("VATNumber") THEN
    CALL "AuditLog"('PERSON',"old_pers"."personid","string"('VAT Number -
',@PersonName),"old_pers"."VATNumber","new_pers"."VATNumber")
  END IF;
  -- VAT Registered
  IF "locate"(@AuditList,',VAT Registered,') > 0 AND UPDATE("VATRegistered")
THEN
    CALL "AuditLog"('PERSON',"old_pers"."personid","string"('VAT Registered
- ',@PersonName),"old_pers"."VATRegistered","new_pers"."VATRegistered")
  END IF;
  -- Company Name
  IF "locate"(@AuditList,',Company Name,') > 0 AND UPDATE("CompanyName")
THEN
    CALL "AuditLog"('PERSON',"old_pers"."personid","string"('Company Name -
',@PersonName),"old_pers"."CompanyName","new_pers"."CompanyName")
  END IF;
  -- Company Registration Number
  IF "locate"(@AuditList,',Company Registration Number,') > 0 AND
UPDATE("RegNo") THEN
    CALL "AuditLog"('PERSON',"old_pers"."personid","string"('Company
Registration Number - ',@PersonName),"old_pers"."RegNo","new_pers"."RegNo")
  END IF;
  -- External Ref Number
  IF "locate"(@AuditList,',External Ref Number,') > 0 AND
UPDATE("ExtNumber") THEN
```

```
    CALL "AuditLog"('PERSON',"old_pers"."personid","string"('External Ref
Number - ',@PersonName),"old_pers"."ExtNumber","new_pers"."ExtNumber")
  END IF;
  -- HMRC Engagement Details
  IF "locate"(@AuditList,',HMRC Engagement Details,') > 0 AND
UPDATE("HMRCEngagementDetail") THEN
    CALL "AuditLog"('PERSON',"old_pers"."personid","string"('HMRC Engagement
Details -
',@PersonName),"old_pers"."HMRCEngagementDetail","new_pers"."HMRCEngagementD
etail")
  END IF;
  -- Bank Account Details
  IF "locate"(@AuditList,',Bank Account Details,') >
AND(UPDATE("BankSortCode") OR UPDATE("BankAcNo") OR UPDATE("BankAcName") OR
UPDATE("BankRef")) THEN
    CALL "AuditLog"('PERSON',"old_pers"."personid","string"('Bank Details -
',@PersonName),"string"('Sort Code ',"old_pers"."BankSortCode",', Acc No.
',"old_pers"."BankAcNo",', Acc Name ',"old_pers"."BankAcName",', Ref.
',"old_pers"."BankRef"),"string"('Sort Code ',"new_pers"."BankSortCode",',
Acc No. ',"new_pers"."BankAcNo",', Acc Name ',"new_pers"."BankAcName",',
Ref. ',"new_pers"."BankRef"))
  END IF;
  -- CompositeID
  IF "locate"(@AuditList,',Composite Pay Company,') >
AND(UPDATE("CompositeID")) THEN
    CALL "AuditLog"('PERSON',"old_pers"."personid","string"('Composite Pay
Company - ',@PersonName),(SELECT "templatename" FROM "composite" AS "c"
WHERE "c"."compositeid" = "old_pers"."compositeid"),(SELECT "templatename"
FROM "composite" AS "c" WHERE "c"."compositeid" = "new_pers"."compositeid"))
  END IF;
  -- SupplierCode
  IF "locate"(@AuditList,',Supplier Code,') > 0 AND(UPDATE("SupplierCode"))
THEN
    CALL "AuditLog"('PERSON',"old_pers"."personid","string"('Supplier Code -
',@PersonName),"old_pers"."SupplierCode","new_pers"."SupplierCode")
  END IF;
  -- CIS (New Style, post 2007)
  IF "locate"(@AuditList,',CIS (post 2007),') >
AND(UPDATE("CISVerificationNumber") OR UPDATE("CISVerificationDate") OR
UPDATE("UniqueTaxReference") OR UPDATE("CISTaxTreatment")) OR
UPDATE("CISEffectiveDate") THEN
    CALL "AuditLog"('PERSON',"old_pers"."personid","string"('CIS Details -
',@PersonName),"string"('Unique Tax Reference
',"old_pers"."UniqueTaxReference",', CIS Verification Number
',"old_pers"."CISVerificationNumber",', CIS Verification Date
',"old_pers"."CISVerificationDate",', CIS Tax Treatment
',"old_pers"."CISTaxTreatment",', CIS Effective Date
',"old_pers"."CISEffectiveDate"),"string"('Unique Tax Reference
```

```
',"new_pers"."UniqueTaxReference",', CIS Verification Number
',"new_pers"."CISVerificationNumber",', CIS Verification Date
',"new_pers"."CISVerificationDate",', CIS Tax Treatment
',"new_pers"."CISTaxTreatment",', CIS Effective Date
',"new_pers"."CISEffectiveDate"))
  END IF
END
GO


COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."Pay_Employee"."PayEmployeeAudit" IS
{CREATE TRIGGER PayEmployeeAudit
 BEFORE UPDATE OF PayMethod,
CompositeID,TaxCode,TaxMethod,NILetter,VATNumber,VATRegistered,CompanyName,R
egNo,ExtNumber,BankSortCode,BankAcNo,Suppliercode,
BankAcName,BankRef,UniqueTaxReference,CISTaxTreatment,CISVerificationDate,CI
SVerificationNumber,CISEffectiveDate,HMRCEngagementDetail ORDER 3 ON
pears.Pay_Employee
REFERENCING OLD AS old_pers NEW AS new_pers
FOR each ROW
WHEN(EXISTS(SELECT * FROM AuditItems WHERE AreaName = 'Person Pay' AND
AuditFlag = 1))
BEGIN
  DECLARE @AuditList long VARCHAR;
  DECLARE @OldDescrip CHAR(250);
  DECLARE @NewDescrip CHAR(250);
  DECLARE @PersonName CHAR(250);
  SELECT string(',',list(ItemName),',') INTO @AuditList FROM AuditItems
WHERE AreaName = 'Person Pay' AND AuditFlag = 1;
  SET @PersonName=(SELECT Name FROM Person WHERE PersonID =
isnull(old_pers.PersonID,new_pers.PersonID));
  -- Payment Method
  IF locate(@AuditList,',Payment Method,') > 0 AND UPDATE(PayMethod) THEN
    CALL AuditLog('PERSON',old_pers.personid,string('Payment Method -
',@PersonName),string(old_pers.PayMethod,CASE old_pers.PayMethod WHEN 1 THEN
'- BACS' WHEN 2 THEN '- Cash' WHEN 3 THEN '- Cheque' ELSE '- Other'
    END),string(new_pers.PayMethod,CASE new_pers.PayMethod WHEN 1 THEN '-
BACS' WHEN 2 THEN '- Cash' WHEN 3 THEN '- Cheque' ELSE '- Other'
    -- Tax Code
    END))
  END IF;
  IF locate(@AuditList,',Tax Code,') > 0 AND UPDATE(TaxCode) THEN
    CALL AuditLog('PERSON',old_pers.personid,string('Tax Code -
',@PersonName),old_pers.TaxCode,new_pers.TaxCode)
  END IF;
  -- Tax Method
  IF locate(@AuditList,',Tax Method,') > 0 AND UPDATE(TaxMethod) THEN
```

```
    CALL AuditLog('PERSON',old_pers.personid,string('Tax Method -
',@PersonName),string(old_pers.TaxMethod,CASE old_pers.TaxMethod WHEN 1 THEN
'- PAYES' WHEN 2 THEN '- Ltd Co' WHEN 3 THEN '- SelfEmp' ELSE '- Other'
    END),string(new_pers.TaxMethod,CASE new_pers.TaxMethod WHEN 1 THEN '-
PAYES' WHEN 2 THEN '- Ltd Co' WHEN 3 THEN '- SelfEmp' ELSE '- Other'
    -- NI Letter
    END))
  END IF;
  IF locate(@AuditList,',NI Letter,') > 0 AND UPDATE(NILetter) THEN
    CALL AuditLog('PERSON',old_pers.personid,string('NI Letter -
',@PersonName),old_pers.NILetter,new_pers.NILetter)
  END IF;
  -- VAT Number
  IF locate(@AuditList,',VAT Number,') > 0 AND UPDATE(VATNumber) THEN
    CALL AuditLog('PERSON',old_pers.personid,string('VAT Number -
',@PersonName),old_pers.VATNumber,new_pers.VATNumber)
  END IF;
  -- VAT Registered
  IF locate(@AuditList,',VAT Registered,') > 0 AND UPDATE(VATRegistered)
THEN
    CALL AuditLog('PERSON',old_pers.personid,string('VAT Registered -
',@PersonName),old_pers.VATRegistered,new_pers.VATRegistered)
  END IF;
  -- Company Name
  IF locate(@AuditList,',Company Name,') > 0 AND UPDATE(CompanyName) THEN
    CALL AuditLog('PERSON',old_pers.personid,string('Company Name -
',@PersonName),old_pers.CompanyName,new_pers.CompanyName)
  END IF;
  -- Company Registration Number
  IF locate(@AuditList,',Company Registration Number,') > 0 AND
UPDATE(RegNo) THEN
    CALL AuditLog('PERSON',old_pers.personid,string('Company Registration
Number - ',@PersonName),old_pers.RegNo,new_pers.RegNo)
  END IF;
  -- External Ref Number
  IF locate(@AuditList,',External Ref Number,') > 0 AND UPDATE(ExtNumber)
THEN
    CALL AuditLog('PERSON',old_pers.personid,string('External Ref Number -
',@PersonName),old_pers.ExtNumber,new_pers.ExtNumber)
  END IF;
  -- HMRC Engagement Details
  IF locate(@AuditList,',HMRC Engagement Details,') > 0 AND
UPDATE(HMRCEngagementDetail) THEN
    CALL AuditLog('PERSON',old_pers.personid,string('HMRC Engagement Details
-
',@PersonName),old_pers.HMRCEngagementDetail,new_pers.HMRCEngagementDetail)
  END IF;
  -- Bank Account Details
```

```
  IF locate(@AuditList,',Bank Account Details,') >
AND(UPDATE(BankSortCode) OR UPDATE(BankAcNo) OR UPDATE(BankAcName) OR
UPDATE(BankRef)) THEN
    CALL AuditLog('PERSON',old_pers.personid,string('Bank Details -
',@PersonName),string('Sort Code ',old_pers.BankSortCode,', Acc No.
',old_pers.BankAcNo,', Acc Name ',old_pers.BankAcName,', Ref.
',old_pers.BankRef),string('Sort Code ',new_pers.BankSortCode,', Acc No.
',new_pers.BankAcNo,', Acc Name ',new_pers.BankAcName,', Ref.
',new_pers.BankRef))
  END IF;
  -- CompositeID
  IF locate(@AuditList,',Composite Pay Company,') >
AND(UPDATE(CompositeID) ) THEN
    CALL AuditLog('PERSON',old_pers.personid,string('Composite Pay Company -
',@PersonName),(SELECT templatename FROM composite c WHERE c.compositeid
=old_pers.compositeid),(SELECT templatename FROM composite c WHERE
c.compositeid =new_pers.compositeid) )
  END IF;
  -- SupplierCode
  IF locate(@AuditList,',Supplier Code,') > 0 AND(UPDATE(SupplierCode) )
THEN
    CALL AuditLog('PERSON',old_pers.personid,string('Supplier Code -
',@PersonName),old_pers.SupplierCode,new_pers.SupplierCode )
  END IF;
  -- CIS (New Style, post 2007)
  IF locate(@AuditList,',CIS (post 2007),') >
AND(UPDATE(CISVerificationNumber) OR UPDATE(CISVerificationDate) OR
UPDATE(UniqueTaxReference) OR UPDATE(CISTaxTreatment)) OR
UPDATE(CISEffectiveDate) THEN
    CALL AuditLog('PERSON',old_pers.personid,string('CIS Details -
',@PersonName),string('Unique Tax Reference ',old_pers.UniqueTaxReference,',
CIS Verification Number ',old_pers.CISVerificationNumber,', CIS Verification
Date ',old_pers.CISVerificationDate,', CIS Tax Treatment
',old_pers.CISTaxTreatment,', CIS Effective Date
',old_pers.CISEffectiveDate),string('Unique Tax Reference
',new_pers.UniqueTaxReference,', CIS Verification Number
',new_pers.CISVerificationNumber,', CIS Verification Date
',new_pers.CISVerificationDate,', CIS Tax Treatment
',new_pers.CISTaxTreatment,', CIS Effective Date
',new_pers.CISEffectiveDate))
  END IF
END
}
GO
```

From:
https://iqxusers.co.uk/iqxhelp/ - **iqx**

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

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