



pears.TempTimeSheet



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

Description

Timesheet headers.

Columns

Column	Type	Null	Default	Comment
TempTimeSheetID	char(20)	NOT NULL		
PlacementID	char(20)	NULL		
SerialNumber	char(12)	NULL		
Period	integer	NOT NULL		
PeriodLength	char(1)	NOT NULL		Week or Month
PayrollIdentifier	char(1)	NULL		For external payroll link
AnalysisCode	char(20)	NULL		
CorrectionFlag	smallint	NULL	0	Non zero if this was to correct errors
Billed	smallint	NULL	0	
Paid	smallint	NULL	0	
PersonID	char(20)	NULL		
TempDeskID	char(20)	NULL		
TransferBatch	integer	NULL		
PayrollNumber	char(15)	NULL		In case employee payroll number changes
TheirRef	char(50)	NULL		
Description	long varchar	NULL		
Currency	char(3)	NULL		
EStatus	smallint	NULL	0	1: E-Timesheet completed on line 2: E-Timesheet changed local copy 3: E-Timesheet awaiting confirmation
StartDate	date	NULL		
EndDate	date	NULL		
ExternalTimesheetReference	char(20)	NULL		



Column	Type	Null	Default	Comment
ExternalInvoiceReference	char(20)	NULL		
SecondaryAgencyID	char(20)	NULL		CompanyID of Secondary Agency which supplied the worker
WhenEntered	timestamp	NULL	current timestamp	
StaffID	char(20)	NULL		
ExternalInvoiceDate	date	NULL		
ExternalInvoiceReceivedDate	date	NULL		
AccordTransferBatch	integer	NULL		
AccordPayrollNumber	char(15)	NULL		In case employee payroll number changes
ScanPathFilename	char(250)	NULL		
TSBarcode	char(50)	NULL		
WhenPaid	timestamp	NULL		
TaxMethod	smallint	NULL		
HolidayPayStatus	smallint	NULL		
Orientation	smallint	NULL		
AWRWasInvolved	smallint	NULL		
ExternalStatus	char(1)	NULL		
VacExtraHols	double	NULL		
ExpenseBenefitCompleted	smallint	NULL	0	
Pay_EmploymentID	char(20)	NULL		Current HMRC data when completed
SupplierCode	char(12)	NULL		Self bill account for secondary agency or temp
SelfBillInvoiceID	char(8)	NULL		
TimesheetCancelReasonID	char(20)	NULL		
ImageBlobID	char(20)	NULL		
NotTaxable	smallint	NULL	0	Flag to indicate off payroll
AnalysisBatch	integer	NULL		
ImageApproved	tinyint	NOT NULL	0	
InvoicePrefix	char(10)	NULL		
InvoiceAddress	long varchar	NULL		
InvoiceEmail	char(250)	NULL		

Primary Key

- TempTimeSheetID



Foreign Keys

Constraint	Columns	References	Delete/update action
Placement	PlacementID	pears.Placement (placementid)	
Person	PersonID	pears.Person (personid)	
TempDesk	TempDeskID	pears.tempdesk (tempdeskid)	
Company	SecondaryAgencyID	pears.Company (companyid)	
Staff	StaffID	pears.staff (staffid)	ON DELETE SET NULL
TimesheetCancelReason	TimesheetCancelReasonID	pears.TimesheetCancelReason (TimesheetCancelReasonID)	
Pay_Employment	Pay_EmploymentID	pears.Pay_Employment (Pay_EmploymentID)	ON DELETE SET NULL

Referenced By

Table	Constraint	Columns	Referenced columns
pears.AWRWeeklyDetail	TempTimesheet	TempTimesheetID	TempTimeSheetID
pears.PDFExtraDocument	TempTimesheet	TempTimesheetID	TempTimeSheetID
pears.TempAccordNIReCharge	TempTimeSheet	TempTimeSheetID	TempTimeSheetID
pears.TempAccordNIReChargeLine	TempTimeSheet	TempTimeSheetID	TempTimeSheetID
pears.TempProvTimeSheetHistory	TempTimesheet	TempTimeSheetID	TempTimeSheetID
pears.TempShift	TempTimesheet	TempTimeSheetID	TempTimeSheetID
pears.TempShiftInvoice	TempTimeSheet	TempTimeSheetID	TempTimeSheetID
pears.TempTimeSheetLine	TempTimeSheet	TempTimeSheetID	TempTimeSheetID
pears.TimesheetSignature	temptimesheet	TemptimesheetID	TempTimeSheetID
pears.TSQueryLog	TempTimeSheet	TempTimeSheetID	TempTimeSheetID

Indexes

Name	Type	Columns	Detail
TempTimeSheet_Period	Index	Period	
TempTimeSheet_transferbatch	Index	TransferBatch	
TempTimeSheet_PerdPID	Index	Period, PersonID	
TempTimeSheet_SerialNumber	Index	SerialNumber	
TempTimeSheet_EndDate	Index	EndDate	
TempTimeSheet_WhenEntered	Index	WhenEntered	
TempTimeSheet_ImageBlobID	Index	ImageBlobID	
TempTimeSheet_ScanPathFilename	Index	ScanPathFilename	
TempTimeSheet_Billed	Index	Billed	
TempTimeSheet_Paid	Index	Paid	



Triggers

Name	Timing	Event
Timesheet_Insert	before	insert order 1
Timesheet_Insert_Corrections	before	insert order 5
TempTimeSheet_update	before	update order 1
Timesheet_Insert_CopyDocs	after	insert order 6

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."TempTimeSheet"
-- Table comment: Timesheet headers.
-- Statement count: 37

CREATE TABLE "pears"."TempTimeSheet" (
    "TempTimeSheetID"          CHAR(20) NOT NULL
    , "PlacementID"             CHAR(20) NULL
    , "SerialNumber"            CHAR(12) NULL
    , "Period"                  INTEGER NOT NULL
    , "PeriodLength"            CHAR(1) NOT NULL
    , "PayrollIdentifier"        CHAR(1) NULL
    , "AnalysisCode"            CHAR(20) NULL
    , "CorrectionFlag"           SMALLINT NULL DEFAULT 0
    , "Billed"                  SMALLINT NULL DEFAULT 0
    , "Paid"                    SMALLINT NULL DEFAULT 0
    , "PersonID"                CHAR(20) NULL
    , "TempDeskID"              CHAR(20) NULL
    , "TransferBatch"           INTEGER NULL
    , "PayrollNumber"           CHAR(15) NULL
    , "TheirRef"                CHAR(50) NULL
    , "Description"              long VARCHAR NULL
    , "Currency"                CHAR(3) NULL
    , "EStatus"                 SMALLINT NULL DEFAULT 0
    , "StartDate"               DATE NULL
    , "EndDate"                 DATE NULL
    , "ExternalTimesheetReference" CHAR(20) NULL
    , "ExternalInvoiceReference" CHAR(20) NULL
    , "SecondaryAgencyID"       CHAR(20) NULL
    , "WhenEntered"             TIMESTAMP NULL DEFAULT CURRENT
TIMESTAMP
    , "StaffID"                 CHAR(20) NULL
    , "ExternalInvoiceDate"      DATE NULL
    , "ExternalInvoiceReceivedDate" DATE NULL
    , "AccordTransferBatch"      INTEGER NULL
```



```
, "AccordPayrollNumber"          CHAR(15) NULL
, "ScanPathFilename"            CHAR(250) NULL
, "TSBarcode"                   CHAR(50) NULL
, "WhenPaid"                    TIMESTAMP NULL
, "TaxMethod"                   SMALLINT NULL
, "HolidayPayStatus"            SMALLINT NULL
, "Orientation"                 SMALLINT NULL
, "AWRWasInvolved"              SMALLINT NULL
, "ExternalStatus"              CHAR(1) NULL
, "VacExtraHols"                DOUBLE NULL
, "ExpenseBenefitCompleted"     SMALLINT NULL DEFAULT 0
, "Pay_EmploymentID"            CHAR(20) NULL
, "SupplierCode"                CHAR(12) NULL
, "SelfBillInvoiceID"           CHAR(8) NULL
, "TimesheetCancelReasonID"     CHAR(20) NULL
, "ImageBlobID"                 CHAR(20) NULL
, "NotTaxable"                  SMALLINT NULL DEFAULT 0
, "AnalysisBatch"               INTEGER NULL
, "ImageApproved"               tinyint NOT NULL DEFAULT 0
, "InvoicePrefix"               CHAR(10) NULL
, "InvoiceAddress"              long VARCHAR NULL
, "InvoiceEmail"                CHAR(250) NULL
, PRIMARY KEY ("TempTimeSheetID" ASC)
, CONSTRAINT "InvalidPeriod" CHECK(("PeriodLength" = 'W' AND "Period"
REGEXP '^(?:19|20)[0-9][0-9](?:[0-0][1-9]|[1-4][0-9]|5[0-3])$')
OR("PeriodLength" = 'M' AND "Period" REGEXP
'^(?:19|20)[0-9][0-9](?:[0-0][1-9]|[1][1-2])$'))
)
GO

COMMENT ON COLUMN "pears"."TempTimeSheet"."PeriodLength" IS
    'Week or Month'
GO

COMMENT ON COLUMN "pears"."TempTimeSheet"."PayrollIdentifier" IS
    'For external payroll link'
GO

COMMENT ON COLUMN "pears"."TempTimeSheet"."CorrectionFlag" IS
    'Non zero if this was to correct errors'
GO

COMMENT ON COLUMN "pears"."TempTimeSheet"."PayrollNumber" IS
    'In case employee payroll number changes'
```



GO

```
COMMENT ON COLUMN "pears"."TempTimeSheet"."EStatus" IS
    '1: E-Timesheet completed on line 2: E-Timesheet changed local copy 3:
    E-Timesheet awaiting confirmation'
```

GO

```
COMMENT ON COLUMN "pears"."TempTimeSheet"."SecondaryAgencyID" IS
    'CompanyID of Secondary Agency which supplied the worker'
```

GO

```
COMMENT ON COLUMN "pears"."TempTimeSheet"."AccordPayrollNumber" IS
    'In case employee payroll number changes'
```

GO

```
COMMENT ON COLUMN "pears"."TempTimeSheet"."Pay_EmploymentID" IS
    'Current HMRC data when completed'
```

GO

```
COMMENT ON COLUMN "pears"."TempTimeSheet"."SupplierCode" IS
    'Self bill account for secondary agency or temp'
```

GO

```
COMMENT ON COLUMN "pears"."TempTimeSheet"."NotTaxable" IS
    'Flag to indicate off payroll'
```

GO

```
COMMENT ON TABLE "pears"."TempTimeSheet" IS
    'Timesheet headers.'
```

GO

```
ALTER TABLE "pears"."TempTimeSheet"
    ADD FOREIGN KEY "Placement" ("PlacementID" ASC)
    REFERENCES "pears"."Placement" ("placementid")
```

GO

```
ALTER TABLE "pears"."TempTimeSheet"
    ADD FOREIGN KEY "Person" ("PersonID" ASC)
    REFERENCES "pears"."Person" ("personid")
```



GO

```
ALTER TABLE "pears"."TempTimeSheet"  
  ADD FOREIGN KEY "TempDesk" ("TempDeskID" ASC)  
  REFERENCES "pears"."tempdesk" ("tempdeskid")
```

GO

```
ALTER TABLE "pears"."TempTimeSheet"  
  ADD FOREIGN KEY "Company" ("SecondaryAgencyID" ASC)  
  REFERENCES "pears"."Company" ("companyid")
```

GO

```
ALTER TABLE "pears"."TempTimeSheet"  
  ADD FOREIGN KEY "Staff" ("StaffID" ASC)  
  REFERENCES "pears"."staff" ("staffid")  
  ON DELETE SET NULL
```

GO

```
ALTER TABLE "pears"."TempTimeSheet"  
  ADD FOREIGN KEY "TimesheetCancelReason" ("TimesheetCancelReasonID" ASC)  
  REFERENCES "pears"."TimesheetCancelReason" ("TimesheetCancelReasonID")
```

GO

```
ALTER TABLE "pears"."TempTimeSheet"  
  ADD FOREIGN KEY "Pay_Employment" ("Pay_EmploymentID" ASC)  
  REFERENCES "pears"."Pay_Employment" ("Pay_EmploymentID")  
  ON DELETE SET NULL
```

GO

```
CREATE INDEX "TempTimeSheet_Period" ON "pears"."TempTimeSheet"  
  ( "Period" )
```

GO

```
CREATE INDEX "TempTimeSheet_transferbatch" ON "pears"."TempTimeSheet"  
  ( "TransferBatch" )
```

GO

```
CREATE INDEX "TempTimeSheet_PerdPID" ON "pears"."TempTimeSheet"  
  ( "Period", "PersonID" )
```

GO



```
CREATE INDEX "TempTimeSheet_SerialNumber" ON "pears"."TempTimeSheet"
  ( "SerialNumber" )
GO

CREATE INDEX "TempTimeSheet_EndDate" ON "pears"."TempTimeSheet"
  ( "EndDate" )
GO

CREATE INDEX "TempTimeSheet_WhenEntered" ON "pears"."TempTimeSheet"
  ( "WhenEntered" )
GO

CREATE INDEX "TempTimeSheet_ImageBlobID" ON "pears"."TempTimeSheet"
  ( "ImageBlobID" )
GO

CREATE INDEX "TempTimeSheet_ScanPathFilename" ON "pears"."TempTimeSheet"
  ( "ScanPathFilename" )
GO

CREATE INDEX "TempTimeSheet_Billed" ON "pears"."TempTimeSheet"
  ( "Billed" )
GO

CREATE INDEX "TempTimeSheet_Paid" ON "pears"."TempTimeSheet"
  ( "Paid" )
GO

CREATE TRIGGER "Timesheet_Insert" BEFORE INSERT ORDER 1 ON
"pears"."TempTimesheet"
REFERENCING NEW AS "new_ts"
FOR each ROW
--when(new_ts.staffid is null)
BEGIN
  SET "new_ts"."serialnumber" = "trim"("new_ts"."serialnumber");
  SET "new_ts"."theirref" = "trim"("new_ts"."theirref");
  IF "new_ts"."taxmethod" IS NULL THEN SET "new_ts"."taxmethod" = (SELECT
"taxmethod" FROM "pay_employee" WHERE "personid" = "new_ts"."personid")
  END IF;
```



```
-- if new_ts.holidaypaystatus is null then set new_ts.holidaypaystatus
--      =(select
isnull(tempholidaycalc(new_ts.personid,truncnum(new_ts.period/100,0),remaind
er(new_ts.period,100),-1),0))
-- end if;
IF "new_ts"."staffid" IS NULL THEN SET "new_ts"."staffid" = "userstaffid"
END IF
exception
  WHEN others THEN SET "new_ts"."staffid" = NULL
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."TempTimeSheet"."Timesheet_Insert" IS
{CREATE TRIGGER Timesheet_Insert
  BEFORE INSERT ORDER 1 ON
pears.TempTimesheet
REFERENCING NEW AS new_ts
FOR each ROW
--when(new_ts.staffid is null)
BEGIN
  SET new_ts.serialnumber = TRIM(new_ts.serialnumber);
  SET new_ts.theirref = TRIM(new_ts.theirref);
  IF new_ts.taxmethod IS NULL THEN SET new_ts.taxmethod=(SELECT taxmethod
FROM pay_employee WHERE personid = new_ts.personid)
  END IF;
-- if new_ts.holidaypaystatus is null then set new_ts.holidaypaystatus
--      =(select
isnull(tempholidaycalc(new_ts.personid,truncnum(new_ts.period/100,0),remaind
er(new_ts.period,100),-1),0))
-- end if;
  IF new_ts.staffid IS NULL THEN SET new_ts.staffid=userstaffid
  END IF
exception
  WHEN others THEN
    SET new_ts.staffid=NULL
END
}
GO

CREATE TRIGGER "Timesheet_Insert_Corrections" BEFORE INSERT ORDER 5 ON
"pears"."TempTimesheet"
REFERENCING NEW AS "new_ts"
FOR each ROW
WHEN("new_ts"."CorrectionFlag" > 0)
BEGIN
```



```
-- Mechanism to mark replacement timesheets paid or billed if only charge
or pay corrected to prevent unnecessary credit notes, payslip ammendments
etc
DECLARE @PayCorrected SMALLINT;
DECLARE @ChargeCorrected SMALLINT;
DECLARE @TempTimeSheetCancellationID CHAR(20);
DECLARE @DoNothing SMALLINT;
-- check there is one and only one matching record in
TempTimeSheetCancellation - otherwise do nothing
IF(SELECT "count"("TempTimeSheetCancellationID") FROM
"TempTimesheetCancellation" WHERE "ReplacementProvTimesheetSerialNumber" =
"new_ts"."SerialNumber") = 1 THEN
-- get cancellation record details
SELECT "TempTimeSheetCancellationID","PayCorrected","ChargeCorrected"
INTO @TempTimeSheetCancellationID,@PayCorrected,
@ChargeCorrected FROM "TempTimesheetCancellation" WHERE
"ReplacementProvTimesheetSerialNumber" = "new_ts"."SerialNumber";
-- Set the paid and billed flags
IF @ChargeCorrected = 0 AND @PayCorrected = 1 THEN
SET "new_TS"."Billed" = 1
END IF;
IF @ChargeCorrected = 1 AND @PayCorrected = 0 THEN
SET "new_TS"."Paid" = 1
END IF;
-- Update the TempTimeSheetCancellation record with new IDs time etc
UPDATE "TempTimesheetCancellation"
SET "ReplacementTempTimeSheetID" = "new_ts"."TempTimesheetID",
"ReplacementProvTimesheetSerialNumber" = NULL,
"WhenReplacementCompleted" = CURRENT_TIMESTAMP,
"WhoCompletedReplacement" = "UserStaffID"
WHERE "TempTimeSheetCancellationID" = @TempTimeSheetCancellationID
END IF
exception
WHEN others THEN SET @DoNothing = 1
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."TempTimeSheet"."Timesheet_Insert_Corrections" IS
{CREATE TRIGGER Timesheet_Insert_Corrections
BEFORE INSERT ORDER 5 ON
pears.TempTimesheet
REFERENCING NEW AS new_ts
FOR each ROW
WHEN(new_ts.CorrectionFlag > 0)
BEGIN
-- Mechanism to mark replacement timesheets paid or billed if only charge
```



or pay corrected to prevent unnecessary credit notes, payslip ammendments etc

```
DECLARE @PayCorrected SMALLINT;
DECLARE @ChargeCorrected SMALLINT;
DECLARE @TempTimeSheetCancellationID CHAR(20);
DECLARE @DoNothing SMALLINT;
-- check there is one and only one matching record in
TempTimeSheetCancellation - otherwise do nothing
IF(SELECT COUNT(TempTimeSheetCancellationID) FROM
TempTimesheetCancellation WHERE ReplacementProvTimesheetSerialNumber =
new_ts.SerialNumber) = 1 THEN
-- get cancellation record details
SELECT TempTimeSheetCancellationID, PayCorrected, ChargeCorrected INTO
@TempTimeSheetCancellationID, @PayCorrected,
@ChargeCorrected FROM TempTimesheetCancellation WHERE
ReplacementProvTimesheetSerialNumber = new_ts.SerialNumber;
-- Set the paid and billed flags
IF @ChargeCorrected = 0 AND @PayCorrected = 1 THEN
SET new_TS.Billed=1
END IF;
IF @ChargeCorrected = 1 AND @PayCorrected = 0 THEN
SET new_TS.Paid=1
END IF;
-- Update the TempTimeSheetCancellation record with new IDs time etc
UPDATE TempTimesheetCancellation SET
ReplacementTempTimeSheetID = new_ts.TempTimesheetID,
ReplacementProvTimesheetSerialNumber = NULL,
WhenReplacementCompleted = CURRENT_TIMESTAMP,
WhoCompletedReplacement = UserStaffID WHERE
TempTimeSheetCancellationID = @TempTimeSheetCancellationID
END IF
exception
WHEN others THEN
SET @DoNothing=1
END
}
GO
```

```
CREATE TRIGGER "TempTimeSheet_update" BEFORE UPDATE ORDER 1 ON
"pears"."TempTimeSheet"
REFERENCING NEW AS "new_ts"
FOR each ROW
BEGIN
SET "new_ts"."serialnumber" = "trim"("new_ts"."serialnumber");
SET "new_ts"."theirref" = "trim"("new_ts"."theirref")
END
GO
```



```
COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."TempTimeSheet"."TempTimeSheet_update" IS
{CREATE TRIGGER TempTimeSheet_update
  BEFORE UPDATE ORDER 1
ON pears.TempTimeSheet
REFERENCING NEW AS new_ts
FOR each ROW
BEGIN
  SET new_ts.serialnumber = TRIM(new_ts.serialnumber);
  SET new_ts.theirref = TRIM(new_ts.theirref)
END
}
GO

CREATE TRIGGER "Timesheet_Insert_CopyDocs" after INSERT ORDER 6 ON
"pears"."TempTimesheet"
REFERENCING NEW AS "new_ts"
FOR each ROW
WHEN("new_ts"."CorrectionFlag" > 0)
BEGIN
  -- Mechanism to mark replacement timesheets paid or billed if only charge
  or pay corrected to prevent unnecessary credit notes, payslip ammendments
  etc
  DECLARE @DoNothing SMALLINT;
  DECLARE @OrigTempTimeSheetID CHAR(20);
  -- check there is one and only one matching record in
  TempTimeSheetCancellation - otherwise do nothing
  IF(SELECT "count"("TempTimeSheetCancellationID") FROM
  "TempTimesheetCancellation" WHERE "Replacementtemptimesheetid" =
  "new_ts"."temptimesheetid") = 1 THEN
    -- get cancellation record details
    SELECT "OriginalTempTimesheetID"
    INTO @OrigTempTimeSheetID FROM "TempTimesheetCancellation" WHERE
    "Replacementtemptimesheetid" = "new_ts"."temptimesheetid";
    INSERT INTO "PDFEXtraDocument"(
    "FilePath", "FileType", "PDFExtraDocumentID", "TempTimesheetID", "PDFImageBl
    obID" )
    SELECT
    "FilePath", "FileType", "uniquekey"("PDFExtraDocumentID"), "new_ts"."TempTi
    mesheetID", "PDFImageBlobID"
    FROM "PDFEXtraDocument" WHERE "TempTimesheetID" =
    @OrigTempTimeSheetID
  END IF
exception
  WHEN others THEN SET @DoNothing = 1
```



```
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."TempTimeSheet"."Timesheet_Insert_CopyDocs" IS
{CREATE TRIGGER Timesheet_Insert_CopyDocs
  after INSERT ORDER 6 ON
pears.TempTimesheet
REFERENCING NEW AS new_ts
FOR each ROW
WHEN(new_ts.CorrectionFlag > 0)
BEGIN
  -- Mechanism to mark replacement timesheets paid or billed if only charge
  or pay corrected to prevent unnecessary credit notes, payslip ammendments
  etc
  DECLARE @DoNothing SMALLINT;
  DECLARE @OrigTempTimeSheetID CHAR(20);
  -- check there is one and only one matching record in
  TempTimeSheetCancellation - otherwise do nothing
  IF(SELECT COUNT(TempTimeSheetCancellationID) FROM
TempTimesheetCancellation WHERE Replacementtemptimesheetid =
new_ts.temptimesheetid) = 1 THEN
    -- get cancellation record details
    SELECT OriginalTempTimesheetID INTO
      @OrigTempTimeSheetID FROM TempTimesheetCancellation WHERE
Replacementtemptimesheetid = new_ts.temptimesheetid;
    INSERT INTO PDFExtraDocument (FileNamePath, FileType,
PDFExtraDocumentID, TempTimesheetID, PDFImageBlobID)
      SELECT FileNamePath, FileType, uniquekey(PDFExtraDocumentID),
new_ts.TempTimesheetID, PDFImageBlobID
      FROM PDFExtraDocument WHERE TempTimesheetID = @OrigTempTimeSheetID
    END IF
  exception
    WHEN others THEN
      SET @DoNothing=1
END
}
GO
```

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

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

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



