



pears.ReferenceRequest



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

Description

Individual Reference Request records

Columns

Column	Type	Null	Default	Comment
ReferenceRequestID	char(20)	NOT NULL		
PersonID	char(20)	NOT NULL		
DepartmentID	char(2)	NOT NULL		
RefereeAddressName	char(60)	NOT NULL		
RefereeSalutation	char(60)	NOT NULL		
RefereePosition	char(40)	NOT NULL		
CompanyName	char(60)	NULL		
Keyname	char(60)	NOT NULL		
Addr1	char(40)	NULL		
Addr2	char(40)	NULL		
Addr3	char(40)	NULL		
Town	char(30)	NULL		
County	char(30)	NULL		
Country	char(30)	NULL		
PostCode	char(20)	NULL		
HomePhone	char(100)	NULL		
WorkPhone	char(100)	NULL		
MobilePhone	char(100)	NULL		
Fax	char(100)	NULL		
Email	char(250)	NOT NULL		
Position	char(100)	NOT NULL		
PositionNotes	long varchar	NULL		
URLRequestCode	char(250)	NULL		
StaffID	char(20)	NULL		
WhenCreated	timestamp	NULL	current timestamp	
WhenCompleted	timestamp	NULL		



Column	Type	Null	Default	Comment
WhenProcessed	timestamp	NULL		
WhoProcessed	char(20)	NULL		
WhenSent	timestamp	NULL		
WhenReturned	timestamp	NULL		
EmploymentID	char(20)	NULL		
RefType	smallint	NULL		
StartDate	timestamp	NULL		
EndDate	timestamp	NULL		
brandid	char(20)	NULL		
ReferenceConfirmationRequired	tinyint	NULL		
NotificationTemplateID	char(50)	NULL		
ReferenceMasterID	char(20)	NULL		
ReminderLastSent	timestamp	NULL		
ReferenceRefused	timestamp	NULL		
ApprovedToSend	tinyint	NULL	0	
Blocked	tinyint	NULL	0	
AskCandidateBeforeSend	tinyint	NULL	0	

Primary Key

- ReferenceRequestID

Foreign Keys

Constraint	Columns	References	Delete/update action
Person	PersonID	pears.Person (personid)	NOT NULL; ON DELETE CASCADE
department	DepartmentID	pears.Department (departmentid)	NOT NULL;
staff	StaffID	pears.staff (staffid)	
NotificationTemplate	NotificationTemplateID	pears.NotificationTemplate (NotificationTemplateID)	ON DELETE SET NULL
ReferenceMaster	ReferenceMasterID	pears.ReferenceMaster (ReferenceMasterID)	ON DELETE SET NULL

Referenced By

- No incoming foreign keys found.



Indexes

Name	Type	Columns	Detail
ReferenceRequest_URLRequestCode	Index	URLRequestCode	

Triggers

Name	Timing	Event
ReferenceRequestUpdate	after	update of "WhenCompleted", "WhenSent", "WhenReturned", "RefereeAddressName", "companyname", "Addr1", "Addr2", "Addr3", "Town", "County", "Country", "Postcode", "Homephone", "mobilephone", "workphone", "fax", "email", "position", "positionnotes" order 3501
ReferenceRequestBlocked	after	update of "Blocked" order 1

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."ReferenceRequest"
-- Table comment: Individual Reference Request records
-- Statement count: 12
```

```
CREATE TABLE "pears"."ReferenceRequest" (
  "ReferenceRequestID"          CHAR(20) NOT NULL
, "PersonID"                    CHAR(20) NOT NULL
, "DepartmentID"                CHAR(2)  NOT NULL
, "RefereeAddressName"          CHAR(60) NOT NULL
, "RefereeSalutation"           CHAR(60) NOT NULL
, "RefereePosition"             CHAR(40) NOT NULL
, "CompanyName"                 CHAR(60)  NULL
, "Keyname"                     CHAR(60) NOT NULL
, "Addr1"                       CHAR(40)  NULL
, "Addr2"                       CHAR(40)  NULL
, "Addr3"                       CHAR(40)  NULL
, "Town"                        CHAR(30)  NULL
, "County"                      CHAR(30)  NULL
, "Country"                     CHAR(30)  NULL
, "PostCode"                    CHAR(20)  NULL
, "HomePhone"                   CHAR(100) NULL
, "WorkPhone"                   CHAR(100) NULL
, "MobilePhone"                 CHAR(100) NULL
, "Fax"                         CHAR(100) NULL
, "Email"                       CHAR(250) NOT NULL
, "Position"                    CHAR(100) NOT NULL
, "PositionNotes"               long VARCHAR NULL
, "URLRequestCode"              CHAR(250) NULL
, "StaffID"                     CHAR(20)  NULL
, "WhenCreated"                 TIMESTAMP NULL DEFAULT CURRENT
TIMESTAMP
```



```
, "WhenCompleted"          TIMESTAMP NULL
, "WhenProcessed"           TIMESTAMP NULL
, "WhoProcessed"            CHAR(20) NULL
, "WhenSent"                TIMESTAMP NULL
, "WhenReturned"            TIMESTAMP NULL
, "EmploymentID"            CHAR(20) NULL
, "RefType"                  SMALLINT NULL
, "StartDate"                TIMESTAMP NULL
, "EndDate"                  TIMESTAMP NULL
, "brandid"                  CHAR(20) NULL
, "ReferenceConfirmationRequired" tinyint NULL
, "NotificationTemplateID"    CHAR(50) NULL
, "ReferenceMasterID"         CHAR(20) NULL
, "ReminderLastSent"          TIMESTAMP NULL
, "ReferenceRefused"          TIMESTAMP NULL
, "ApprovedToSend"           tinyint NULL DEFAULT 0
, "Blocked"                   tinyint NULL DEFAULT 0
, "AskCandidateBeforeSend"    tinyint NULL DEFAULT 0
, PRIMARY KEY ("ReferenceRequestID" ASC)
)
GO

COMMENT ON TABLE "pears"."ReferenceRequest" IS
'Individual Reference Request records'
GO

ALTER TABLE "pears"."ReferenceRequest"
  ADD NOT NULL FOREIGN KEY "Person" ("PersonID" ASC)
  REFERENCES "pears"."Person" ("personid")
  ON DELETE CASCADE
GO

ALTER TABLE "pears"."ReferenceRequest"
  ADD NOT NULL FOREIGN KEY "department" ("DepartmentID" ASC)
  REFERENCES "pears"."Department" ("departmentid")
GO

ALTER TABLE "pears"."ReferenceRequest"
  ADD FOREIGN KEY "staff" ("StaffID" ASC)
  REFERENCES "pears"."staff" ("staffid")
GO

ALTER TABLE "pears"."ReferenceRequest"
```



```
ADD FOREIGN KEY "NotificationTemplate" ("NotificationTemplateID" ASC)
REFERENCES "pears"."NotificationTemplate" ("NotificationTemplateID")
ON DELETE SET NULL
GO

ALTER TABLE "pears"."ReferenceRequest"
ADD FOREIGN KEY "ReferenceMaster" ("ReferenceMasterID" ASC)
REFERENCES "pears"."ReferenceMaster" ("ReferenceMasterID")
ON DELETE SET NULL
GO

CREATE INDEX "ReferenceRequest_URLRequestCode" ON "pears"."ReferenceRequest"
( "URLRequestCode" )
GO

CREATE TRIGGER "ReferenceRequestUpdate" after UPDATE OF "WhenCompleted",
"WhenSent", "WhenReturned", "RefereeAddressName", "companyname",
"Addr1", "Addr2", "Addr3", "Town", "County", "Country", "Postcode", "Homephone", "mo
bilephone", "workphone", "fax", "email", "position", "positionnotes" ORDER 3501
ON
"pears"."ReferenceRequest"
REFERENCING OLD AS "old_name" NEW AS "new_name"
FOR each ROW
BEGIN
    DECLARE "rv" long VARCHAR;
    SET "rv" = '';
    IF UPDATE("WhenSent") THEN
        CALL "AuditLog"('PERSON',"old_name"."PersonID",
            "string"('Reference Requested -
', "new_name"."RefereeAddressName"),'', "string"("new_name"."whensent"))
        END IF;
    IF UPDATE("WhenReturned") THEN
        CALL "AuditLog"('PERSON',"old_name"."PersonID",
            "string"('Reference Returned -
', "new_name"."RefereeAddressName"),'', "string"("new_name"."whenreturned"))
        END IF;
    IF UPDATE("WhenCompleted") THEN
        CALL "AuditLog"('PERSON',"old_name"."PersonID",
            "string"('Reference Completed -
', "new_name"."RefereeAddressName"),'', "string"("new_name"."whencompleted"))
        END IF;
    IF UPDATE("DepartmentID") THEN SET "rv" = "string"("rv", 'DepartmentID:
', "old_name"."departmentid", ' to
', "new_name"."departmentid", "char"(13), "char"(10)) END IF;
    IF UPDATE("RefereeAddressName") THEN SET "rv" =
```



```
"string"("rv",'RefereeAddressName: ','old_name"."RefereeAddressName",' to
','new_name"."RefereeAddressName","char"(13),"char"(10)) END IF;
  IF UPDATE("companyname") THEN SET "rv" = "string"("rv",'CompanyName:
','old_name"."companyname",' to
','new_name"."companyname","char"(13),"char"(10)) END IF;
  IF UPDATE("RefereePosition") THEN SET "rv" = "string"("rv",'DepartmentID:
','old_name"."departmentid",' to
','new_name"."departmentid","char"(13),"char"(10)) END IF;
  IF UPDATE("Addr1") THEN SET "rv" = "string"("rv",'Addr1:
','old_name"."Addr1",' to ','new_name"."Addr1","char"(13),"char"(10)) END
IF;
  IF UPDATE("Addr2") THEN SET "rv" = "string"("rv",'Addr2:
','old_name"."Addr2",' to ','new_name"."Addr2","char"(13),"char"(10)) END
IF;
  IF UPDATE("Addr3") THEN SET "rv" = "string"("rv",'Addr3:
','old_name"."Addr3",' to ','new_name"."Addr3","char"(13),"char"(10)) END
IF;
  IF UPDATE("Town") THEN SET "rv" = "string"("rv",'Town:
','old_name"."Town",' to ','new_name"."Town","char"(13),"char"(10)) END IF;
  IF UPDATE("County") THEN SET "rv" = "string"("rv",'County:
','old_name"."County",' to ','new_name"."County","char"(13),"char"(10)) END
IF;
  IF UPDATE("Country") THEN SET "rv" = "string"("rv",'Country:
','old_name"."Country",' to ','new_name"."Country","char"(13),"char"(10))
END IF;
  IF UPDATE("PostCode") THEN SET "rv" = "string"("rv",'PostCode:
','old_name"."PostCode",' to ','new_name"."PostCode","char"(13),"char"(10))
END IF;
  IF UPDATE("HomePhone") THEN SET "rv" = "string"("rv",'HomePhone:
','old_name"."HomePhone",' to
','new_name"."HomePhone","char"(13),"char"(10)) END IF;
  IF UPDATE("WorkPhone") THEN SET "rv" = "string"("rv",'WorkPhone:
','old_name"."WorkPhone",' to
','new_name"."WorkPhone","char"(13),"char"(10)) END IF;
  IF UPDATE("MobilePhone") THEN SET "rv" = "string"("rv",'MobilePhone:
','old_name"."MobilePhone",' to
','new_name"."MobilePhone","char"(13),"char"(10)) END IF;
  IF UPDATE("Fax") THEN SET "rv" = "string"("rv",'Fax: ','old_name"."Fax",'
to ','new_name"."Fax","char"(13),"char"(10)) END IF;
  IF UPDATE("Email") THEN SET "rv" = "string"("rv",'Email:
','old_name"."Email",' to ','new_name"."Email","char"(13),"char"(10)) END
IF;
  IF UPDATE("Position") THEN SET "rv" = "string"("rv",'Position:
','old_name"."Position",' to ','new_name"."Position","char"(13),"char"(10))
END IF;
  IF UPDATE("PositionNotes") THEN SET "rv" = "string"("rv",'PositionNotes:
','old_name"."PositionNotes",' to
','new_name"."PositionNotes","char"(13),"char"(10)) END IF;
```



```
IF "rv" <> '' THEN
    CALL "AuditLog"('PERSON',"old_name"."personid","string"('Reference
Update - ', "new_name"."RefereeAddressName"),'', "rv")
END IF
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."ReferenceRequest"."ReferenceRequestUpdate" IS
{CREATE TRIGGER ReferenceRequestUpdate
after UPDATE OF WhenCompleted, WhenSent,
WhenReturned, RefereeAddressName, companyname,
Addr1, Addr2, Addr3, Town, County, Country, Postcode, Homephone, mobilephone, workpho
ne, fax, email, POSITION, positionnotes ORDER 3501 ON
pears.ReferenceRequest
REFERENCING OLD AS old_name NEW AS new_name
FOR each ROW
BEGIN
DECLARE rv long VARCHAR;
SET rv = '';
IF UPDATE(WhenSent) THEN
    CALL AuditLog('PERSON',old_name.PersonID,
    string('Reference Requested -
',new_name.RefereeAddressName),'',string(new_name.whensent));
END IF;
IF UPDATE(WhenReturned) THEN
    CALL AuditLog('PERSON',old_name.PersonID,
    string('Reference Returned -
',new_name.RefereeAddressName),'',string(new_name.whenreturned));
END IF;
IF UPDATE(WhenCompleted) THEN
    CALL AuditLog('PERSON',old_name.PersonID,
    string('Reference Completed -
',new_name.RefereeAddressName),'',string(new_name.whencompleted));
END IF;
IF UPDATE(DepartmentID) THEN SET rv = string(rv, 'DepartmentID: ',
old_name.departmentid, ' to ', new_name.departmentid, CHAR(13),CHAR(10)) END
IF;
IF UPDATE(RefereeAddressName) THEN SET rv = string(rv, 'RefereeAddressName:
', old_name.RefereeAddressName, ' to ', new_name.RefereeAddressName,
CHAR(13),CHAR(10)) END IF;
IF UPDATE(companyname) THEN SET rv = string(rv, 'CompanyName: ',
old_name.companyname, ' to ', new_name.companyname, CHAR(13),CHAR(10)) END
IF;
IF UPDATE(RefereePosition) THEN SET rv = string(rv, 'DepartmentID: ',
old_name.departmentid, ' to ', new_name.departmentid, CHAR(13),CHAR(10)) END
IF;
```



```
IF UPDATE(Addr1) THEN SET rv =string(rv, 'Addr1: ', old_name.Addr1, ' to ',
new_name.Addr1, CHAR(13),CHAR(10)) END IF;
IF UPDATE(Addr2) THEN SET rv =string(rv, 'Addr2: ', old_name.Addr2, ' to ',
new_name.Addr2, CHAR(13),CHAR(10)) END IF;
IF UPDATE(Addr3) THEN SET rv = string(rv,'Addr3: ', old_name.Addr3, ' to ',
new_name.Addr3, CHAR(13),CHAR(10)) END IF;
IF UPDATE(Town) THEN SET rv = string(rv,'Town: ', old_name.Town, ' to ',
new_name.Town, CHAR(13),CHAR(10)) END IF;
IF UPDATE(County) THEN SET rv = string(rv,'County: ', old_name.County, ' to
', new_name.County, CHAR(13),CHAR(10)) END IF;
IF UPDATE(Country) THEN SET rv = string(rv,'Country: ', old_name.Country, '
to ', new_name.Country, CHAR(13),CHAR(10)) END IF;
IF UPDATE(PostCode) THEN SET rv = string(rv,'PostCode: ',
old_name.PostCode, ' to ', new_name.PostCode, CHAR(13),CHAR(10)) END IF;
IF UPDATE(HomePhone) THEN SET rv = string(rv,'HomePhone: ',
old_name.HomePhone, ' to ', new_name.HomePhone, CHAR(13),CHAR(10)) END IF;
IF UPDATE(WorkPhone) THEN SET rv = string(rv,'WorkPhone: ',
old_name.WorkPhone, ' to ', new_name.WorkPhone, CHAR(13),CHAR(10)) END IF;
IF UPDATE(MobilePhone) THEN SET rv =string(rv, 'MobilePhone: ',
old_name.MobilePhone, ' to ', new_name.MobilePhone, CHAR(13),CHAR(10)) END
IF;
IF UPDATE(Fax) THEN SET rv = string(rv,'Fax: ', old_name.Fax, ' to ',
new_name.Fax, CHAR(13),CHAR(10)) END IF;
IF UPDATE(Email) THEN SET rv = string(rv,'Email: ', old_name.Email, ' to ',
new_name.Email, CHAR(13),CHAR(10)) END IF;
IF UPDATE(POSITION) THEN SET rv = string(rv,'Position: ',
old_name.Position, ' to ', new_name.Position, CHAR(13),CHAR(10)) END IF;
IF UPDATE(PositionNotes) THEN SET rv = string(rv,'PositionNotes: ',
old_name.PositionNotes, ' to ', new_name.PositionNotes, CHAR(13),CHAR(10))
END IF;
IF rv!='' THEN
CALL "AuditLog"('PERSON',"old_name"."personid","string"('Reference Update -
',new_name.RefereeAddressName),'', rv);
END IF;
END
}
GO
```

```
CREATE TRIGGER "ReferenceRequestBlocked" after UPDATE OF "Blocked"
ORDER 1 ON
"pears"."ReferenceRequest"
REFERENCING OLD AS "old_name" NEW AS "new_name"
FOR each ROW
BEGIN
CALL "AuditLog"('PERSON',"old_name"."PersonID",
"string"('Reference Request Blocking -
',"new_name"."RefereeAddressName"),"old_name"."blocked","new_name"."blocked"
```



```
)  
END  
GO  
  
COMMENT TO PRESERVE FORMAT ON TRIGGER  
"pears"."ReferenceRequest"."ReferenceRequestBlocked" IS  
{CREATE TRIGGER ReferenceRequestBlocked  
  after UPDATE OF Blocked ORDER 1 ON  
pears.ReferenceRequest  
REFERENCING OLD AS old_name NEW AS new_name  
FOR each ROW  
BEGIN  
  CALL AuditLog('PERSON',old_name.PersonID,  
    string('Reference Request Blocking -  
,new_name.RefereeAddressName),old_name.blocked,new_name.blocked);  
END  
}  
GO
```

From:

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

Permanent link:

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

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

