



pears.oledocument



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

Description

Pointers to documents belonging to People & Companies. Stored in the Blobstore (Class O)

Columns

Column	Type	Null	Default	Comment
oledocumentid	char(20)	NOT NULL		
description	char(50)	NULL		
ownertype	char(2)	NULL		
ownerid	char(20)	NULL		
RawFile	tinyint	NULL		Set if storing the original file NOT in an OLE wrapper
FileExtension	char(5)	NULL		If RawFile
DocumentTypeID	char(20)	NULL		
ExternalFilePath	long varchar	NULL		
LinkCreatedBy	char(25)	NULL		
LinkCreatedDateTime	timestamp	NULL		
WhenArchived	timestamp	NULL		
WhoArchived	char(20)	NULL		
WhenRedacted	timestamp	NULL		
WhoRedacted	char(20)	NULL		
GlobalDoc	smallint	NULL		
Defunct	tinyint	NULL	0	
deletiondate	date	NULL		
Signed	smallint	NULL		
ExpiryDate	date	NULL		

Primary Key

- oledocumentid



Foreign Keys

Constraint	Columns	References	Delete/update action
DocumentType	DocumentTypeID	pears.DocumentType (DocumentTypeID)	ON DELETE SET NULL

Referenced By

Table	Constraint	Columns	Referenced columns
pears.EmailToSendAttachment	oledocument	OLEDocumentID	oledocumentid

Indexes

Name	Type	Columns	Detail
oledoc_owner	Index	ownertype, ownerid	
oledocumentOwneridDocumentTypeID	Index	ownertype, ownerid, DocumentTypeID	

Triggers

Name	Timing	Event
oledocumentauditinsdel	before	insert,delete order 1
oledocumentautoarchive	after	insert order 2
oledocument_updateaudit	after	update of "deletiondate" order 3

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."oledocument"
-- Table comment: Pointers to documents belonging to People & Companies.
-- Stored in the Blobstore (Class 0)
-- Statement count: 13
```

```
CREATE TABLE "pears"."oledocument" (
  "oledocumentid"          CHAR(20) NOT NULL
, "description"             CHAR(50) NULL
, "ownertype"               CHAR(2) NULL
, "ownerid"                 CHAR(20) NULL
, "RawFile"                 tinyint NULL
, "FileExtension"           CHAR(5) NULL
, "DocumentTypeID"          CHAR(20) NULL
, "ExternalFilePath"        long VARCHAR NULL
, "LinkCreatedBy"           CHAR(25) NULL
, "LinkCreatedDateTime"     TIMESTAMP NULL
, "WhenArchived"            TIMESTAMP NULL
```



```
, "WhoArchived"          CHAR(20) NULL
, "WhenRedacted"         TIMESTAMP NULL
, "WhoRedacted"          CHAR(20) NULL
, "GlobalDoc"            SMALLINT NULL
, "Defunct"              tinyint NULL DEFAULT 0
, "deletiondate"         DATE NULL
, "Signed"               SMALLINT NULL
, "ExpiryDate"           DATE NULL
, PRIMARY KEY ("oledocumentid" ASC)
)
GO

COMMENT ON COLUMN "pears"."oledocument"."RawFile" IS
    'Set if storing the original file NOT in an OLE wrapper'
GO

COMMENT ON COLUMN "pears"."oledocument"."FileExtension" IS
    'If RawFile'
GO

COMMENT ON TABLE "pears"."oledocument" IS
    'Pointers to documents belonging to People & Companies.  Stored in the
    Blobstore (Class 0)'
GO

ALTER TABLE "pears"."oledocument"
    ADD FOREIGN KEY "DocumentType" ("DocumentTypeID" ASC)
    REFERENCES "pears"."DocumentType" ("DocumentTypeID")
    ON DELETE SET NULL
GO

CREATE INDEX "oledoc_owner" ON "pears"."oledocument"
    ( "ownertype", "ownerid" )
GO

CREATE INDEX "oledocumentOwneridDocumentTypeID" ON "pears"."oledocument"
    ( "ownertype", "ownerid", "DocumentTypeID" )
GO

CREATE TRIGGER "oledocumentauditinsdel" BEFORE INSERT,DELETE ORDER 1 ON
    "pears"."oledocument"
```



```
REFERENCING OLD AS "old_d" NEW AS "new_d"
FOR each ROW
WHEN(EXISTS(SELECT * FROM "AuditItems" WHERE "AreaName" = 'Document' AND
"AuditFlag" = 1))
BEGIN
    DECLARE "OwnerType" CHAR(20);
    DECLARE "OwnerName" CHAR(100);
    IF inserting AND EXISTS(SELECT * FROM "AuditItems" WHERE "AreaName" =
'Document' AND "AuditFlag" = 1 AND "ItemName" = 'Add') THEN
        SELECT CASE "new_d"."ownertype" WHEN 'P' THEN 'Person' WHEN 'C' THEN
'Company' WHEN 'L' THEN 'Placement' WHEN 'V' THEN 'Vacancy' END INTO
"OwnerType";
        SELECT CASE "new_d"."ownertype"
            WHEN 'P' THEN(SELECT "name" FROM "person" WHERE "personid" =
"new_d"."ownerid")
            WHEN 'C' THEN(SELECT "name" FROM "company" WHERE "companyid" =
"new_d"."ownerid")
            WHEN 'L' THEN(SELECT "refcode" FROM "placement" WHERE "placementid" =
"new_d"."ownerid")
            WHEN 'V' THEN(SELECT "string"("Name",' -
', "v"."position", '(' , "v"."RefCode", ')') FROM "Company" KEY JOIN "Employment"
KEY JOIN "vacancy" AS "v" WHERE "vacancyID" = "new_d"."ownerid") END
            INTO "OwnerName";
        CALL "AuditLog"('DOCUMENT',"new_d"."ownerid","string"("OwnerType",'
',"OwnerName",' Document Inserted.'),',',"new_d"."description")
    END IF;
    IF deleting AND EXISTS(SELECT * FROM "AuditItems" WHERE "AreaName" =
'Document' AND "AuditFlag" = 1 AND "ItemName" = 'Remove') THEN
        SELECT CASE "old_d"."ownertype" WHEN 'P' THEN 'Person' WHEN 'C' THEN
'Company' WHEN 'L' THEN 'Placement' WHEN 'V' THEN 'Vacancy' END INTO
"OwnerType";
        SELECT CASE "old_d"."ownertype"
            WHEN 'P' THEN(SELECT "name" FROM "person" WHERE "personid" =
"old_d"."ownerid")
            WHEN 'C' THEN(SELECT "name" FROM "company" WHERE "companyid" =
"old_d"."ownerid")
            WHEN 'L' THEN(SELECT "refcode" FROM "placement" WHERE "placementid" =
"old_d"."ownerid")
            WHEN 'V' THEN(SELECT "string"("Name",' -
', "v"."position", '(' , "v"."RefCode", ')') FROM "Company" KEY JOIN "Employment"
KEY JOIN "vacancy" AS "v" WHERE "vacancyID" = "old_d"."ownerid") END
            INTO "OwnerName";
        CALL "AuditLog"('DOCUMENT',"old_d"."ownerid","string"("OwnerType",'
',"OwnerName",' Document Deleted. '), "old_d"."description",',')
    END IF
END
GO
```



```
COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."oledocument"."oledocumentauditinsdel" IS
{CREATE TRIGGER oledocumentauditinsdel
  BEFORE INSERT,DELETE ORDER 1 ON
pears.oledocument
REFERENCING OLD AS old_d NEW AS new_d
FOR each ROW
WHEN(EXISTS(SELECT * FROM AuditItems WHERE AreaName = 'Document' AND
AuditFlag = 1))
BEGIN
  DECLARE OwnerType CHAR(20);
  DECLARE OwnerName CHAR(100);
  IF inserting AND EXISTS(SELECT * FROM AuditItems WHERE AreaName =
'Document' AND AuditFlag = 1 AND ItemName = 'Add') THEN
    SELECT CASE new_d.ownertype WHEN 'P' THEN 'Person' WHEN 'C' THEN
'Company' WHEN 'L' THEN 'Placement' WHEN 'V' THEN 'Vacancy' END INTO
OwnerType;
    SELECT CASE new_d.ownertype
      WHEN 'P' THEN (SELECT name FROM person WHERE personid = new_d.ownerid)
      WHEN 'C' THEN (SELECT name FROM company WHERE companyid =
new_d.ownerid)
      WHEN 'L' THEN (SELECT refcode FROM placement WHERE placementid =
new_d.ownerid)
      WHEN 'V' THEN (SELECT string(Name, ' - ', v.position, '(', v.RefCode, ')'))
FROM Company KEY JOIN Employment KEY JOIN vacancy v WHERE vacancyID =
new_d.ownerid)
    END INTO OwnerName;
    CALL AuditLog('DOCUMENT',new_d.ownerid,string(OwnerType, ' ',OwnerName, '
Document Inserted. '),'',new_d.description)
  END IF;
  IF deleting AND EXISTS(SELECT * FROM AuditItems WHERE AreaName =
'Document' AND AuditFlag = 1 AND ItemName = 'Remove') THEN
    SELECT CASE old_d.ownertype WHEN 'P' THEN 'Person' WHEN 'C' THEN
'Company' WHEN 'L' THEN 'Placement' WHEN 'V' THEN 'Vacancy' END INTO
OwnerType;
    SELECT CASE old_d.ownertype
      WHEN 'P' THEN (SELECT name FROM person WHERE personid = old_d.ownerid)
      WHEN 'C' THEN (SELECT name FROM company WHERE companyid =
old_d.ownerid)
      WHEN 'L' THEN (SELECT refcode FROM placement WHERE placementid =
old_d.ownerid)
      WHEN 'V' THEN (SELECT string(Name, ' - ', v.position, '(', v.RefCode, ')'))
FROM Company KEY JOIN Employment KEY JOIN vacancy v WHERE vacancyID =
old_d.ownerid)
    END INTO OwnerName;
    CALL AuditLog('DOCUMENT',old_d.ownerid,string(OwnerType, ' ',OwnerName, '
Document Deleted. '),old_d.description, '')
```



```
END IF
END
}
GO

CREATE TRIGGER "oledocumentautoarchive" after INSERT ORDER 2 ON
"pears"."oledocument"
REFERENCING NEW AS "new_d"
FOR each ROW
BEGIN
  DECLARE "staffid" CHAR(20);
  IF "varexists"('userstaffid') <> 0 THEN
    SET "staffid" = "userstaffid"
  ELSE SET "staffid" = NULL
  END IF;
  IF EXISTS(SELECT * FROM "documenttype" WHERE "isnull"("iscompliance",0) =
1 AND "documenttypeid" = "new_d"."documenttypeid") THEN
    IF "new_d"."whenredacted" IS NOT NULL THEN
      UPDATE "oledocument" SET "whenarchived" = CURRENT
TIMESTAMP,"whoarchived" = "staffid" WHERE "whenredacted" IS NOT NULL AND
"whenarchived" IS NULL AND "documenttypeid" = "new_d"."documenttypeid"
      AND "oledocumentid" <> "new_d"."oledocumentid" AND "ownerid" =
"new_d"."ownerid" AND "ownertype" = "new_d"."ownertype"
    END IF;
    IF "new_d"."whenredacted" IS NULL THEN
      UPDATE "oledocument" SET "whenarchived" = CURRENT
TIMESTAMP,"whoarchived" = "staffid" WHERE "whenredacted" IS NULL AND
"whenarchived" IS NULL AND "documenttypeid" = "new_d"."documenttypeid"
      AND "oledocumentid" <> "new_d"."oledocumentid" AND "ownerid" =
"new_d"."ownerid" AND "ownertype" = "new_d"."ownertype"
    END IF
  END IF
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."oledocument"."oledocumentautoarchive" IS
{CREATE TRIGGER oledocumentautoarchive
after INSERT ORDER 2 ON
pears.oledocument
REFERENCING NEW AS new_d
FOR each ROW
BEGIN
  DECLARE staffid CHAR(20);
  IF varexists('userstaffid') <> 0 THEN
    SET staffid=userstaffid
```



```
ELSE SET staffid=NULL
END IF;
IF EXISTS(SELECT * FROM documenttype WHERE isnull(iscompliance,0) = 1 AND
documenttypeid = new_d.documenttypeid) THEN
    IF new_d.whenredacted IS NOT NULL THEN
        UPDATE oledocument SET whenarchived = CURRENT_TIMESTAMP,whoarchived =
staffid WHERE whenredacted IS NOT NULL AND whenarchived IS NULL AND
documenttypeid = new_d.documenttypeid
        AND oledocumentid <> new_d.oledocumentid AND ownerid = new_d.ownerid
AND ownertype=new_d.ownertype
    END IF;
    IF new_d.whenredacted IS NULL THEN
        UPDATE oledocument SET whenarchived = CURRENT_TIMESTAMP,whoarchived =
staffid WHERE whenredacted IS NULL AND whenarchived IS NULL AND
documenttypeid = new_d.documenttypeid
        AND oledocumentid <> new_d.oledocumentid AND ownerid = new_d.ownerid
AND ownertype=new_d.ownertype
    END IF
END IF
END
}
GO
```

```
CREATE TRIGGER "oledocument_updateaudit" after UPDATE OF "deletiondate"
ORDER 3 ON "pears"."oledocument"
REFERENCING OLD AS "old_d" NEW AS "new_d"
FOR each ROW
WHEN(EXISTS(SELECT * FROM "AuditItems" WHERE "AreaName" = 'Document' AND
"AuditFlag" = 1))
BEGIN
    DECLARE "OwnerType" CHAR(20);
    DECLARE "OwnerName" CHAR(100);
    IF EXISTS(SELECT * FROM "AuditItems" WHERE "AreaName" = 'Document' AND
"AuditFlag" = 1 AND "ItemName" = 'Deletion Date') THEN
        SELECT CASE "old_d"."ownertype" WHEN 'P' THEN 'Person' WHEN 'C' THEN
'Company' WHEN 'L' THEN 'Placement' WHEN 'V' THEN 'Vacancy' END INTO
"OwnerType";
        SELECT CASE "old_d"."ownertype"
            WHEN 'P' THEN(SELECT "name" FROM "person" WHERE "personid" =
"old_d"."ownerid")
            WHEN 'C' THEN(SELECT "name" FROM "company" WHERE "companyid" =
"old_d"."ownerid")
            WHEN 'L' THEN(SELECT "refcode" FROM "placement" WHERE "placementid" =
"old_d"."ownerid")
            WHEN 'V' THEN(SELECT "string"("Name",' -
','v"."position",'(',"v"."RefCode",')') FROM "Company" KEY JOIN "Employment"
KEY JOIN "vacancy" AS "v" WHERE "vacancyID" = "old_d"."ownerid") END
```



```
        INTO "OwnerName";
        CALL "AuditLog"('DOCUMENT',"old_d"."ownerid","string"("OwnerType",'
',"OwnerName",' - ','old_d"."description",' Deletion Date Changed.
'),"old_d"."deletiondate","new_d"."deletiondate")
    END IF
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."oledocument"."oledocument_updateaudit" IS
{CREATE TRIGGER oledocument_updateaudit
after UPDATE OF deletiondate
ORDER 3 ON pears.oledocument
REFERENCING OLD AS old_d NEW AS new_d
FOR each ROW
WHEN(EXISTS(SELECT * FROM AuditItems WHERE AreaName = 'Document' AND
AuditFlag = 1))
BEGIN
    DECLARE OwnerType CHAR(20);
    DECLARE OwnerName CHAR(100);
    IF EXISTS(SELECT * FROM AuditItems WHERE AreaName = 'Document' AND
AuditFlag = 1 AND ItemName = 'Deletion Date') THEN
        SELECT CASE old_d.ownertype WHEN 'P' THEN 'Person' WHEN 'C' THEN 'Company'
WHEN 'L' THEN 'Placement' WHEN 'V' THEN 'Vacancy' END INTO OwnerType;
        SELECT CASE old_d.ownertype
            WHEN 'P' THEN (SELECT name FROM person WHERE personid = old_d.ownerid)
            WHEN 'C' THEN (SELECT name FROM company WHERE companyid =
old_d.ownerid)
            WHEN 'L' THEN (SELECT refcode FROM placement WHERE placementid =
old_d.ownerid)
            WHEN 'V' THEN (SELECT string(Name,' - ',v.position,'(',v.RefCode,')'))
FROM Company KEY JOIN Employment KEY JOIN vacancy v WHERE vacancyID =
old_d.ownerid)
        END INTO OwnerName;
        CALL AuditLog('DOCUMENT',old_d.ownerid,string(OwnerType,' ',OwnerName,' -
',old_d.description,' Deletion Date Changed.
'),old_d.deletiondate,new_d.deletiondate)
    END IF
END
}
GO
```



From:

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

Permanent link:

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

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

