



# pears.blobstore



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

## Description

OLE objects and other blob data eg pictures. If external Blob Store is in use the records are links to the relevant files.

## Columns

| Column            | Type        | Null     | Default | Comment                                                                                             |
|-------------------|-------------|----------|---------|-----------------------------------------------------------------------------------------------------|
| class             | char(1)     | NOT NULL |         |                                                                                                     |
| id                | char(20)    | NOT NULL |         | class of :- J or V: PersonID, O: OleDocumentID, M: MailMergeID, T: TemplateStore, L: ContactEventID |
| blob              | long binary | NULL     |         |                                                                                                     |
| createdby         | char(15)    | NULL     |         |                                                                                                     |
| createdat         | timestamp   | NULL     |         |                                                                                                     |
| changedby         | char(15)    | NULL     |         |                                                                                                     |
| changedat         | timestamp   | NULL     |         |                                                                                                     |
| accessedby        | char(15)    | NULL     |         |                                                                                                     |
| accessedat        | timestamp   | NULL     |         |                                                                                                     |
| zipped            | smallint    | NULL     | 0       |                                                                                                     |
| PublishToWeb      | smallint    | NULL     | 0       |                                                                                                     |
| AddToInvPDF       | smallint    | NULL     | 0       |                                                                                                     |
| CVSendOutApproved | smallint    | NULL     | 0       |                                                                                                     |
| ExternalFilePath  | char(250)   | NULL     |         | If null blob is stored internally in the blob column                                                |
| Optimised         | timestamp   | NULL     |         | Date blob was last optimised. Cleared by trigger if CreateDat or ChangeDat are updated.             |
| filesize          | integer     | NULL     |         | Kb. Internal and external, disc not unzipped size                                                   |

## Primary Key

- class, id



## Foreign Keys

- No outgoing foreign keys found.

## Referenced By

- No incoming foreign keys found.

## Indexes

| Name                 | Type  | Columns          | Detail |
|----------------------|-------|------------------|--------|
| blobcreatedby        | Index | createdby        |        |
| blobchangedby        | Index | changedby        |        |
| blobaccessedby       | Index | accessedby       |        |
| blobaccessedat       | Index | accessedat       |        |
| BlobExternalFilePath | Index | ExternalFilePath |        |

## Triggers

| Name                             | Timing | Event                                      |
|----------------------------------|--------|--------------------------------------------|
| blobstore_delete                 | before | delete order 1                             |
| BlobStore_ClearOptimisedDateTime | before | update of "createdat", "changedat" order 1 |
| blobstore_auditphoto             | after  | insert,delete,update order 10              |

## Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."blobstore"
-- Table comment: OLE objects and other blob data eg pictures. If external
Blob Store is in use the records are links to the relevant files.
-- Statement count: 18
```

```
CREATE TABLE "pears"."blobstore" (
  "class" CHAR(1) NOT NULL
, "id" CHAR(20) NOT NULL
, "blob" long BINARY NULL
, "createdby" CHAR(15) NULL
, "createdat" TIMESTAMP NULL
, "changedby" CHAR(15) NULL
, "changedat" TIMESTAMP NULL
, "accessedby" CHAR(15) NULL
```



```
, "accessedat"          TIMESTAMP NULL
, "zipped"              SMALLINT NULL DEFAULT 0
, "PublishToWeb"        SMALLINT NULL DEFAULT 0
, "AddToInvPDF"          SMALLINT NULL DEFAULT 0
, "CVSendOutApproved"   SMALLINT NULL DEFAULT 0
, "ExternalFilePath"     CHAR(250) NULL
, "Optimised"            TIMESTAMP NULL
, "filesize"             INTEGER NULL
, PRIMARY KEY ("class" ASC, "id" ASC)
)
GO

COMMENT ON COLUMN "pears"."blobstore"."id" IS
    'class of :- J or V: PersonID, O: OleDocumentID, M: MailMergeID, T:
    TemplateStore, L: ContactEventID'
GO

COMMENT ON COLUMN "pears"."blobstore"."ExternalFilePath" IS
    'If null blob is stored internally in the blob column'
GO

COMMENT ON COLUMN "pears"."blobstore"."Optimised" IS
    'Date blob was last optimised. Cleared by trigger if CreateDat or
    ChangeDat are updated.'
GO

COMMENT ON COLUMN "pears"."blobstore"."filesize" IS
    'Kb. Internal and external, disc not unzipped size'
GO

COMMENT ON TABLE "pears"."blobstore" IS
    'OLE objects and other blob data eg pictures. If external Blob Store is
    in use the records are links to the relevant files.'
GO

CREATE INDEX "blobcreatedby" ON "pears"."blobstore"
    ( "createdby" )
GO

CREATE INDEX "blobchangedby" ON "pears"."blobstore"
    ( "changedby" )
```



GO

```
CREATE INDEX "blobaccessedby" ON "pears"."blobstore"  
  ( "accessedby" )
```

GO

```
CREATE INDEX "blobaccessedat" ON "pears"."blobstore"  
  ( "accessedat" )
```

GO

```
CREATE INDEX "BlobExternalFilePath" ON "pears"."blobstore"  
  ( "ExternalFilePath" )
```

GO

```
CREATE TRIGGER "blobstore_delete" BEFORE DELETE ORDER 1 ON  
"pears"."blobstore"  
REFERENCING OLD AS "old_blob"  
FOR each ROW  
WHEN("old_blob"."externalfilepath" IS NOT NULL)  
BEGIN  
  DECLARE "CurrentUser" CHAR(25);  
  IF "VAREXISTS"('UserStaffID') = 1 THEN  
    SET "CurrentUser" = "UserStaffID"  
  END IF;  
  INSERT INTO "BlobStoreDeleteQueue"(  
"Class", "ID", "WhoDeleted", "ExternalFilePath" ) VALUES(  
"old_blob"."Class", "old_blob"."ID", "CurrentUser", "old_blob"."ExternalFilePat  
h" )  
END  
GO
```

```
COMMENT TO PRESERVE FORMAT ON TRIGGER "pears"."blobstore"."blobstore_delete"  
IS  
{CREATE TRIGGER blobstore_delete  
  BEFORE DELETE ORDER 1 ON  
"pears"."blobstore"  
REFERENCING OLD AS "old_blob"  
FOR each ROW  
WHEN("old_blob"."externalfilepath" IS NOT NULL)  
BEGIN  
  DECLARE CurrentUser CHAR(25);  
  IF VAREXISTS('UserStaffID') = 1 THEN  
    SET CurrentUser = UserStaffID;
```



```
END IF;
INSERT INTO BlobStoreDeleteQueue(Class, ID, WhoDeleted, ExternalFilePath)
VALUES(old_blob.Class, old_blob.ID, CurrentUser, old_blob.ExternalFilePath)
END
}
GO

CREATE TRIGGER "BlobStore_ClearOptimisedDateTime" BEFORE UPDATE OF
"createdat",
"changedat" ORDER 1 ON "pears"."blobstore"
REFERENCING NEW AS "new_Blob"
FOR each ROW
BEGIN
SET "new_Blob"."Optimised" = NULL
END
GO

COMMENT ON TRIGGER "pears"."blobstore"."BlobStore_ClearOptimisedDateTime" IS
'Clears the Optimised datetime column if either the CreateDat or
ChangeDat fields are updated - indicating that the Blob needs to be examined
again.'
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."blobstore"."BlobStore_ClearOptimisedDateTime" IS
{CREATE TRIGGER BlobStore_ClearOptimisedDateTime
BEFORE UPDATE OF createdat,
changedat ORDER 1 ON pears.blobstore
REFERENCING NEW AS new_Blob
FOR each ROW
BEGIN
SET new_Blob.Optimised = NULL
END
}
GO

CREATE TRIGGER "blobstore_auditphoto" after INSERT,DELETE,UPDATE ORDER 10 ON
"pears"."blobstore"
REFERENCING OLD AS "oldrec" NEW AS "newrec"
FOR each ROW WHEN("oldrec"."class" = 'J' OR "newrec"."class" = 'J')
BEGIN
IF inserting THEN
CALL "AuditLog"('PERSON',"newrec"."ID",'Picture Inserted',NULL,'New
Picture')
```



```
ELSEIF deleting THEN
    CALL "AuditLog"('PERSON',"oldrec"."ID",'Picture Deleted','Old
Picture',NULL)
ELSEIF updating THEN
    IF "isnull"("oldrec"."changedat",CURRENT TIMESTAMP) <>
"newrec"."changedat" THEN
        CALL "AuditLog"('PERSON',"oldrec"."ID",'Picture Changed','Old
Picture','New Picture')
    END IF
END IF
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."blobstore"."blobstore_auditphoto" IS
{CREATE TRIGGER blobstore_auditphoto
after INSERT,DELETE,UPDATE ORDER 10 ON
"pears"."blobstore"
REFERENCING OLD AS "oldrec" NEW AS "newrec"
FOR each ROW WHEN("oldrec"."class" = 'J' OR "newrec"."class" = 'J')
BEGIN
    IF inserting THEN
        CALL "AuditLog"('PERSON',"newrec"."ID",'Picture Inserted',NULL,'New
Picture')
    ELSEIF deleting THEN
        CALL "AuditLog"('PERSON',"oldrec"."ID",'Picture Deleted','Old
Picture',NULL)
    ELSEIF updating THEN
        IF "isnull"("oldrec"."changedat",CURRENT TIMESTAMP) <>
"newrec"."changedat" THEN
            CALL "AuditLog"('PERSON',"oldrec"."ID",'Picture Changed','Old
Picture','New Picture')
        END IF END IF
END
}
GO
```

From:

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

Permanent link:

[https://iqxusers.co.uk/iqxhelp/doku.php?id=database:tables:pears\\_blobstore](https://iqxusers.co.uk/iqxhelp/doku.php?id=database:tables:pears_blobstore)

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

