



pears.OffLimits



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

Description

Limits on approaching Companies or People during specified period.

Columns

Column	Type	Null	Default	Comment
OffLimitsID	char(20)	NOT NULL		
PersonID	char(20)	NULL		
CompanyID	char(20)	NULL		
Until	date	NOT NULL		
Reason	char(255)	NULL		
StaffID	char(20)	NOT NULL		
WhenEntered	timestamp	NOT NULL	current timestamp	

Primary Key

- OffLimitsID

Foreign Keys

Constraint	Columns	References	Delete/update action
Person	PersonID	pears.Person (personid)	ON DELETE CASCADE
Company	CompanyID	pears.Company (companyid)	ON DELETE CASCADE
Staff	StaffID	pears.staff (staffid)	NOT NULL;

Referenced By

- No incoming foreign keys found.



Indexes

Name	Type	Columns	Detail
OffLimits_CompanyIDUntil	Index	CompanyID, Until	
OffLimits_PersonIDUntil	Index	PersonID, Until	

Triggers

Name	Timing	Event
AuditOffLimits	before	insert,delete,update order 1

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."OffLimits"
-- Table comment: Limits on approaching Companies or People during specfied
period.
-- Statement count: 9
```

```
CREATE TABLE "pears"."OffLimits" (
    "OffLimitsID"          CHAR(20) NOT NULL
    , "PersonID"           CHAR(20) NULL
    , "CompanyID"          CHAR(20) NULL
    , "Until"              DATE NOT NULL
    , "Reason"             CHAR(255) NULL
    , "StaffID"            CHAR(20) NOT NULL
    , "WhenEntered"        TIMESTAMP NOT NULL DEFAULT CURRENT
```

```
TIMESTAMP
    , PRIMARY KEY ("OffLimitsID" ASC)
)
GO
```

```
COMMENT ON TABLE "pears"."OffLimits" IS
    'Limits on approaching Companies or People during specfied period.'
GO
```

```
ALTER TABLE "pears"."OffLimits"
    ADD FOREIGN KEY "Person" ("PersonID" ASC)
    REFERENCES "pears"."Person" ("personid")
    ON DELETE CASCADE
GO
```



```
ALTER TABLE "pears"."OffLimits"
  ADD FOREIGN KEY "Company" ("CompanyID" ASC)
  REFERENCES "pears"."Company" ("companyid")
  ON DELETE CASCADE
GO

ALTER TABLE "pears"."OffLimits"
  ADD NOT NULL FOREIGN KEY "Staff" ("StaffID" ASC)
  REFERENCES "pears"."staff" ("staffid")
GO

CREATE INDEX "OffLimits_CompanyIDUntil" ON "pears"."OffLimits"
  ( "CompanyID","Until" DESC )
GO

CREATE INDEX "OffLimits_PersonIDUntil" ON "pears"."OffLimits"
  ( "PersonID","Until" DESC )
GO

CREATE TRIGGER "AuditOffLimits" BEFORE INSERT,DELETE,UPDATE ORDER 1 ON
"pears"."OffLimits"
REFERENCING OLD AS "OldOffLimits" NEW AS "NewOffLimits"
FOR each ROW
BEGIN
  IF inserting THEN
    CASE
      WHEN "NewOffLimits"."CompanyID" IS NOT NULL THEN
        CALL "AuditLog"('COMPANY',"NewOffLimits"."CompanyID",'Off Limits
Restriction Inserted',NULL,"string"("NewOffLimits"."Reason",' -
',"NewOffLimits"."Until"))
      WHEN "NewOffLimits"."PersonID" IS NOT NULL THEN
        CALL "AuditLog"('PERSON',"NewOffLimits"."PersonID",'Off Limits
Restriction Inserted',NULL,"string"("NewOffLimits"."Reason",' -
',"NewOffLimits"."Until"))
      END CASE
    ELSEIF deleting THEN
      CASE
        WHEN "OldOffLimits"."CompanyID" IS NOT NULL THEN
          CALL "AuditLog"('COMPANY',"OldOffLimits"."CompanyID",'Off Limits
Restriction Deleted',"string"("OldOffLimits"."Reason",' -
',"OldOffLimits"."Until"),NULL)
        WHEN "NewOffLimits"."PersonID" IS NOT NULL THEN
          CALL "AuditLog"('PERSON',"OldOffLimits"."PersonID",'Off Limits
```



```
Restriction Deleted','"string"("OldOffLimits"."Reason",' -
','"OldOffLimits"."Until"),NULL)
    END CASE
ELSEIF updating THEN
    CASE
    WHEN "OldOffLimits"."CompanyID" IS NOT NULL THEN
        CALL "AuditLog"('COMPANY',"OldOffLimits"."CompanyID",'Off Limits
Restriction Updated','"string"("OldOffLimits"."Reason",' -
','"OldOffLimits"."Until"),"string"("NewOffLimits"."Reason",' -
','"NewOffLimits"."Until"))
    WHEN "NewOffLimits"."PersonID" IS NOT NULL THEN
        CALL "AuditLog"('PERSON',"OldOffLimits"."PersonID",'Off Limits
Restriction Updated','"string"("OldOffLimits"."Reason",' -
','"OldOffLimits"."Until"),"string"("NewOffLimits"."Reason",' -
','"NewOffLimits"."Until"))
    END CASE
END IF
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER "pears"."OffLimits"."AuditOffLimits"
IS
{CREATE TRIGGER AuditOffLimits
    BEFORE INSERT,DELETE,UPDATE ORDER 1 ON
pears.OffLimits
REFERENCING OLD AS OldOffLimits NEW AS NewOffLimits
FOR each ROW
BEGIN
    IF inserting THEN
        CASE
        WHEN NewOffLimits.CompanyID IS NOT NULL THEN
            CALL AuditLog('COMPANY',NewOffLimits.CompanyID,'Off Limits Restriction
Inserted',NULL,string(NewOffLimits.Reason,' - ',NewOffLimits.Until))
        WHEN NewOffLimits.PersonID IS NOT NULL THEN
            CALL AuditLog('PERSON',NewOffLimits.PersonID,'Off Limits Restriction
Inserted',NULL,string(NewOffLimits.Reason,' - ',NewOffLimits.Until))
        END CASE
    ELSEIF deleting THEN
        CASE
        WHEN OldOffLimits.CompanyID IS NOT NULL THEN
            CALL AuditLog('COMPANY',OldOffLimits.CompanyID,'Off Limits Restriction
Deleted',string(OldOffLimits.Reason,' - ',OldOffLimits.Until),NULL)
        WHEN NewOffLimits.PersonID IS NOT NULL THEN
            CALL AuditLog('PERSON',OldOffLimits.PersonID,'Off Limits Restriction
Deleted',string(OldOffLimits.Reason,' - ',OldOffLimits.Until),NULL)
        END CASE
    ELSEIF updating THEN
```



```
CASE
  WHEN OldOffLimits.CompanyID IS NOT NULL THEN
    CALL AuditLog('COMPANY',OldOffLimits.CompanyID,'Off Limits Restriction
Updated',string(OldOffLimits.Reason,' - ',NewOffLimits.Until))
  WHEN NewOffLimits.PersonID IS NOT NULL THEN
    CALL AuditLog('PERSON',OldOffLimits.PersonID,'Off Limits Restriction
Updated',string(OldOffLimits.Reason,' - ',NewOffLimits.Until))
  END CASE
END IF
END
}
GO
```

From:

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

Permanent link:

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

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

