pears.DivisionAccess

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

Description

Links Staff records to Divisions. Staff with no records have access to all Divisions.

Columns

Column	Type	Null	Default	Comment
divisionid	char(20)	NOT NULL		
staffid	char(20)	NOT NULL		

Primary Key

- divisionid, staffid

Foreign Keys

Constraint	Columns	References	Delete/update action
division	divisionid	pears.Division (divisionid)	NOT NULL; ON DELETE CASCADE
staff	staffid	pears.staff (staffid)	NOT NULL; ON DELETE CASCADE

Referenced By

- No incoming foreign keys found.

Indexes

- No indexes found.

Triggers

Name	Timing	Event
WPK_divisionaccess_DIVISION	after	insert,delete,update order 1
DivisionAccessInsertAudit	after	insert order 1
DivisionAccessDeleteAudit	after	delete order 1

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."DivisionAccess"
-- Table comment: Links Staff records to Divisions. Staff with no records
have access to all Divisions.
-- Statement count: 10

CREATE TABLE "pears"."DivisionAccess" (
    "divisionid"                    CHAR(20) NOT NULL
   ,"staffid"                       CHAR(20) NOT NULL
   ,PRIMARY KEY ("divisionid" ASC,"staffid" ASC)
)
GO


COMMENT ON TABLE "pears"."DivisionAccess" IS
    'Links Staff records to Divisions. Staff with no records have access to
all Divisions.'
GO


ALTER TABLE "pears"."DivisionAccess"
    ADD NOT NULL FOREIGN KEY "division" ("divisionid" ASC)
    REFERENCES "pears"."Division" ("divisionid")
    ON DELETE CASCADE
GO


ALTER TABLE "pears"."DivisionAccess"
    ADD NOT NULL FOREIGN KEY "staff" ("staffid" ASC)
    REFERENCES "pears"."staff" ("staffid")
    ON DELETE CASCADE
GO


CREATE TRIGGER "WPK_divisionaccess_DIVISION" after INSERT,DELETE,UPDATE
ORDER 1 ON
```

```
"pears"."divisionaccess"
FOR each statement
BEGIN
  CALL "WPKTrackChange"('P','DIVISION')
END
GO


COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."DivisionAccess"."WPK_divisionaccess_DIVISION" IS
{CREATE TRIGGER WPK_divisionaccess_DIVISION
 after INSERT,DELETE,UPDATE ORDER 1 ON
divisionaccess
FOR each statement
BEGIN
  CALL WPKTrackChange('P','DIVISION')
END
}
GO


CREATE TRIGGER "DivisionAccessInsertAudit" after INSERT ORDER 1 ON
"pears"."DivisionAccess"
REFERENCING NEW AS "new_access"
FOR each ROW
WHEN(EXISTS(SELECT * FROM "AuditItems" WHERE "AreaName" = 'User' AND
"AuditFlag" = 1 AND "ItemName" = 'Division Access'))
BEGIN
  DECLARE "RoleName" CHAR(50);
  DECLARE "StaffName" CHAR(25);
  SELECT "name" INTO "RoleName" FROM "division" WHERE "divisionid" =
"new_access"."divisionid";
  SELECT "userid" INTO "StaffName" FROM "staff" WHERE "staffid" =
"new_access"."staffid";
  CALL "AuditLog"('USER','',"string"("StaffName",' Added division
',"RoleName"),'','Added')
END
GO


COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."DivisionAccess"."DivisionAccessInsertAudit" IS
{CREATE TRIGGER DivisionAccessInsertAudit
 after INSERT ORDER 1 ON
DivisionAccess
REFERENCING NEW AS new_access
FOR each ROW
WHEN(EXISTS(SELECT * FROM AuditItems WHERE AreaName = 'User' AND AuditFlag =
```

```
1 AND ItemName = 'Division Access'))
BEGIN
  DECLARE RoleName CHAR(50);
  DECLARE StaffName CHAR(25);
  SELECT name INTO RoleName FROM division WHERE divisionid =
new_access.divisionid;
  SELECT userid INTO StaffName FROM staff WHERE staffid =
new_access.staffid;
  CALL AuditLog('USER','',string(StaffName,' Added division
',RoleName),'','Added')
END
}
GO


CREATE TRIGGER "DivisionAccessDeleteAudit" after DELETE ORDER 1 ON
"pears"."DivisionAccess"
REFERENCING OLD AS "old_access"
FOR each ROW
WHEN(EXISTS(SELECT * FROM "AuditItems" WHERE "AreaName" = 'User' AND
"AuditFlag" = 1 AND "ItemName" = 'Division Access'))
BEGIN
  DECLARE "RoleName" CHAR(50);
  DECLARE "StaffName" CHAR(25);
  SELECT "name" INTO "RoleName" FROM "division" WHERE "divisionid" =
"old_access"."divisionid";
  SELECT "userid" INTO "StaffName" FROM "staff" WHERE "staffid" =
"old_access"."staffid";
  CALL "AuditLog"('USER','',"string"("StaffName",' Removed division
',"RoleName"),'Deleted','')
END
GO


COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."DivisionAccess"."DivisionAccessDeleteAudit" IS
{CREATE TRIGGER DivisionAccessDeleteAudit
 after DELETE ORDER 1 ON
DivisionAccess
REFERENCING OLD AS old_access
FOR each ROW
WHEN(EXISTS(SELECT * FROM AuditItems WHERE AreaName = 'User' AND AuditFlag =
1 AND ItemName = 'Division Access'))
BEGIN
  DECLARE RoleName CHAR(50);
  DECLARE StaffName CHAR(25);
  SELECT name INTO RoleName FROM division WHERE divisionid =
old_access.divisionid;
```

```
  SELECT userid INTO StaffName FROM staff WHERE staffid =
old_access.staffid;
  CALL AuditLog('USER','',string(StaffName,' Removed division
',RoleName),'Deleted','')
END
}
GO
```