



pears.WPKAssignedRole



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

Description

Link between Staff and WPKRoles.

Columns

Column	Type	Null	Default	Comment
WPKRoleID	char(20)	NOT NULL		
OwnerID	char(20)	NOT NULL		
Expires	date	NULL		

Primary Key

- WPKRoleID, OwnerID

Foreign Keys

- No outgoing foreign keys found.

Referenced By

- No incoming foreign keys found.

Indexes

Name	Type	Columns	Detail
RoleOwner	Index	OwnerID	



Triggers

Name	Timing	Event
WPKTrackAssigRoleChange	after	insert,delete,update order 1
WPKAssignedRoleInsertAudit	after	insert order 1
WPKAssignedRoleDeleteAudit	after	delete order 1

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."WPKAssignedRole"
-- Table comment: Link between Staff and WPKRoles.
-- Statement count: 9

CREATE TABLE "pears"."WPKAssignedRole" (
    "WPKRoleID"          CHAR(20) NOT NULL
    , "OwnerID"          CHAR(20) NOT NULL
    , "Expires"          DATE NULL
    , PRIMARY KEY ( "WPKRoleID" ASC, "OwnerID" ASC)
)
GO

COMMENT ON TABLE "pears"."WPKAssignedRole" IS
    'Link between Staff and WPKRoles.'
GO

CREATE INDEX "RoleOwner" ON "pears"."WPKAssignedRole"
    ( "OwnerID" )
GO

CREATE TRIGGER "WPKTrackAssigRoleChange" after INSERT,DELETE,UPDATE ORDER 1
ON
    "pears"."WPKAssignedRole"
FOR each statement
BEGIN
    CALL "WPKTrackChange"('R','*')
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
    "pears"."WPKAssignedRole"."WPKTrackAssigRoleChange" IS
```



```
{CREATE TRIGGER WPKTrackAssigRoleChange
  after INSERT,DELETE,UPDATE ORDER 1 ON
WPKAssignedRole
FOR each statement
BEGIN
  CALL WPKTrackChange('R','*')
END
}
GO

CREATE TRIGGER "WPKAssignedRoleInsertAudit" after INSERT ORDER 1 ON
"pears"."WPKAssignedRole"
REFERENCING NEW AS "new_role"
FOR each ROW
WHEN(EXISTS(SELECT * FROM "AuditItems" WHERE "AreaName" = 'User' AND
"AuditFlag" = 1))
BEGIN
  DECLARE "AuditList" long VARCHAR;
  DECLARE "RoleName" CHAR(50);
  DECLARE "StaffName" CHAR(25);
  SELECT "string"(',' ,',' ,list("ItemName"),',' ,') INTO "AuditList" FROM
"AuditItems" WHERE "AreaName" = 'User' AND "AuditFlag" = 1;
  IF "locate"("AuditList",',Roles,') > 0 AND EXISTS(SELECT * FROM "wpkrole"
WHERE "wpkroleid" = "new_role"."wpkroleid" AND "rolegroup" <> 'USER') AND
"new_role"."wpkroleid" <> 'USERSCRNMOD' THEN
    SELECT "name" INTO "RoleName" FROM "wpkrole" WHERE "wpkroleid" =
"new_role"."wpkroleid";
    SELECT "userid" INTO "StaffName" FROM "staff" WHERE "staffid" =
"new_role"."ownerid";
    CALL "AuditLog"('USER',',',' ,string("StaffName",' Added
',' ,RoleName'),'Added',')
  END IF;
  IF "locate"("AuditList",',Layout Settings,') > 0 AND(EXISTS(SELECT * FROM
"wpkrole" WHERE "wpkroleid" = "new_role"."wpkroleid" AND "rolegroup" =
'USER') OR "new_role"."wpkroleid" = 'USERSCRNMOD') THEN
    SELECT "name" INTO "RoleName" FROM "wpkrole" WHERE "wpkroleid" =
"new_role"."wpkroleid";
    SELECT "userid" INTO "StaffName" FROM "staff" WHERE "staffid" =
"new_role"."ownerid";
    CALL "AuditLog"('USER',',',' ,string("StaffName",' Added
',' ,RoleName'),'Added',')
  END IF
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
```



```
"pears"."WPKAssignedRole"."WPKAssignedRoleInsertAudit" IS
{CREATE TRIGGER WPKAssignedRoleInsertAudit
  after INSERT ORDER 1 ON
WPKAssignedRole
REFERENCING NEW AS new_role
FOR each ROW
WHEN(EXISTS(SELECT * FROM AuditItems WHERE AreaName = 'User' AND AuditFlag =
1))
BEGIN
  DECLARE AuditList long VARCHAR;
  DECLARE RoleName CHAR(50);
  DECLARE StaffName CHAR(25);
  SELECT string(',',list(ItemName),',') INTO AuditList FROM AuditItems WHERE
AreaName = 'User' AND AuditFlag = 1;
  IF locate(AuditList,',Roles,') > 0 AND EXISTS(SELECT * FROM wpkrole WHERE
wpkroleid = new_role.wpkroleid AND rolegroup <> 'USER') AND
new_role.wpkroleid <> 'USERSCRNMOD' THEN
    SELECT name INTO RoleName FROM wpkrole WHERE wpkroleid =
new_role.wpkroleid;
    SELECT userid INTO StaffName FROM staff WHERE staffid =
new_role.ownerid;
    CALL AuditLog('USER','',string(StaffName,' Added ',RoleName),'Added','')
  END IF;
  IF locate(AuditList,',Layout Settings,') > 0 AND (EXISTS(SELECT * FROM
wpkrole WHERE wpkroleid = new_role.wpkroleid AND rolegroup = 'USER') OR
new_role.wpkroleid = 'USERSCRNMOD') THEN
    SELECT name INTO RoleName FROM wpkrole WHERE wpkroleid =
new_role.wpkroleid;
    SELECT userid INTO StaffName FROM staff WHERE staffid =
new_role.ownerid;
    CALL AuditLog('USER','',string(StaffName,' Added ',RoleName),'Added','')
  END IF;
END
}
GO
```

```
CREATE TRIGGER "WPKAssignedRoleDeleteAudit" after DELETE ORDER 1 ON
"pears"."WPKAssignedRole"
REFERENCING OLD AS "old_role"
FOR each ROW
WHEN(EXISTS(SELECT * FROM "AuditItems" WHERE "AreaName" = 'User' AND
"AuditFlag" = 1))
BEGIN
  DECLARE "AuditList" long VARCHAR;
  DECLARE "RoleName" CHAR(50);
  DECLARE "StaffName" CHAR(25);
  SELECT "string"(',',"list"("ItemName"),',') INTO "AuditList" FROM
```



```
"AuditItems" WHERE "AreaName" = 'User' AND "AuditFlag" = 1;
  IF "locate"("AuditList",'Roles,') > 0 AND EXISTS(SELECT * FROM "wpkrole"
WHERE "wpkroleid" = "old_role"."wpkroleid" AND "rolegroup" <> 'USER') AND
"old_role"."wpkroleid" <> 'USERSCRNMOD' THEN
  SELECT "name" INTO "RoleName" FROM "wpkrole" WHERE "wpkroleid" =
"old_role"."wpkroleid";
  SELECT "userid" INTO "StaffName" FROM "staff" WHERE "staffid" =
"old_role"."ownerid";
  CALL "AuditLog"('USER','',"string"("StaffName",' Removed
',"RoleName"),'', 'Removed')
END IF;
  IF "locate"("AuditList",'Layout Settings,') > 0 AND (EXISTS(SELECT * FROM
"wpkrole" WHERE "wpkroleid" = "old_role"."wpkroleid" AND "rolegroup" =
'USER') OR "old_role"."wpkroleid" = 'USERSCRNMOD') THEN
  SELECT "name" INTO "RoleName" FROM "wpkrole" WHERE "wpkroleid" =
"old_role"."wpkroleid";
  SELECT "userid" INTO "StaffName" FROM "staff" WHERE "staffid" =
"old_role"."ownerid";
  CALL "AuditLog"('USER','',"string"("StaffName",' Removed
',"RoleName"),'', 'Removed')
END IF
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."WPKAssignedRole"."WPKAssignedRoleDeleteAudit" IS
{CREATE TRIGGER WPKAssignedRoleDeleteAudit
after DELETE ORDER 1 ON
WPKAssignedRole
REFERENCING OLD AS old_role
FOR each ROW
WHEN(EXISTS(SELECT * FROM AuditItems WHERE AreaName = 'User' AND AuditFlag =
1))
BEGIN
  DECLARE AuditList long VARCHAR;
  DECLARE RoleName CHAR(50);
  DECLARE StaffName CHAR(25);
  SELECT string(',',list(ItemName),',') INTO AuditList FROM AuditItems WHERE
AreaName = 'User' AND AuditFlag = 1;
  IF locate(AuditList,'Roles,') > 0 AND EXISTS(SELECT * FROM wpkrole WHERE
wpkroleid = old_role.wpkroleid AND rolegroup <> 'USER') AND
old_role.wpkroleid <> 'USERSCRNMOD' THEN
    SELECT name INTO RoleName FROM wpkrole WHERE wpkroleid =
old_role.wpkroleid;
    SELECT userid INTO StaffName FROM staff WHERE staffid =
old_role.ownerid;
    CALL AuditLog('USER','',"string(StaffName,' Removed
```



```
' ,RoleName), '', 'Removed')
END IF;
IF locate(AuditList, 'Layout Settings,') > 0 AND (EXISTS(SELECT * FROM
wpkrole WHERE wpkroleid = old_role.wpkroleid AND rolegroup = 'USER') OR
old_role.wpkroleid = 'USERSCRNMOD') THEN
SELECT name INTO RoleName FROM wpkrole WHERE wpkroleid =
old_role.wpkroleid;
SELECT userid INTO StaffName FROM staff WHERE staffid =
old_role.ownerid;
CALL AuditLog('USER', '', string(StaffName, ' Removed
', RoleName), '', 'Removed')
END IF;
END
}
GO
```

From:

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

Permanent link:

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

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

