



pears.AWRvacancy



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

Description

AWR settings for Vacancy records. One-to-one relationship with Vacancy table. Overrides AWRCompany.

Columns

Column	Type	Null	Default	Comment
vacancyid	char(20)	NOT NULL		
AWRStatus	smallint	NOT NULL	1	
pay	long varchar	NULL		
holidays	long varchar	NULL		
benefits	long varchar	NULL		
notes	long varchar	NULL		
bonus	long varchar	NULL		
extrahols	double	NULL		
AWRCheckDate	date	NULL		

Primary Key

- vacancyid

Foreign Keys

Constraint	Columns	References	Delete/update action
vacancy	vacancyid	pears.vacancy (vacancyid)	NOT NULL; ON DELETE CASCADE

Referenced By

- No incoming foreign keys found.



Indexes

- No indexes found.

Triggers

Name	Timing	Event
AWRvacancy_update	after	update of "AWRStatus", "AWRCheckDate" order 1

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."AWRvacancy"
-- Table comment: AWR settings for Vacancy records. One-to-one relationship
with Vacancy table. Overrides AWRCompany.
-- Statement count: 5

CREATE TABLE "pears"."AWRvacancy" (
    "vacancyid"                CHAR(20) NOT NULL
    , "AWRStatus"              SMALLINT NOT NULL DEFAULT 1
    , "pay"                    long VARCHAR NULL
    , "holidays"               long VARCHAR NULL
    , "benefits"                long VARCHAR NULL
    , "notes"                   long VARCHAR NULL
    , "bonus"                   long VARCHAR NULL
    , "extrahols"               DOUBLE NULL
    , "AWRCheckDate"           DATE NULL
    , PRIMARY KEY ("vacancyid" ASC)
)
GO

COMMENT ON TABLE "pears"."AWRvacancy" IS
    'AWR settings for Vacancy records. One-to-one relationship with Vacancy
table. Overrides AWRCompany.'
GO

ALTER TABLE "pears"."AWRvacancy"
    ADD NOT NULL FOREIGN KEY "vacancy" ("vacancyid" ASC)
    REFERENCES "pears"."vacancy" ("vacancyid")
    ON DELETE CASCADE
GO
```



```
CREATE TRIGGER "AWRvacancy_update" after UPDATE OF "AWRStatus",
"AWRCheckDate" ORDER 1 ON "pears"."AWRvacancy"
REFERENCING OLD AS "old_awr" NEW AS "new_awr"
FOR each ROW
BEGIN
    DECLARE @CompName CHAR(250);
    SET @CompName = (SELECT "string"("Company"."Name",' -
', "vacancy"."position",'(' , "vacancy"."RefCode",' )') FROM "Company" KEY JOIN
"Employment" KEY JOIN "vacancy" WHERE "vacancy"."vacancyid" =
"old_awr"."vacancyID");
    -- audit
    IF UPDATE("AWRStatus") THEN
        CALL "AuditLog"('AWRVAC',"old_awr"."vacancyID",
        "string"('AWR Status Vacancy - ',@CompName),CASE "old_awr"."AWRStatus"
WHEN 1 THEN 'Not Known' WHEN 2 THEN 'AWR Applies' WHEN 3 THEN 'AWR not
Applicable' END,
        CASE "new_awr"."AWRStatus" WHEN 1 THEN 'Not Known' WHEN 2 THEN 'AWR
Applies' WHEN 3 THEN 'AWR not Applicable' END)
    END IF;
    IF UPDATE("AWRCheckDate") THEN
        CALL "AuditLog"('AWRVAC',"old_awr"."vacancyID",
        "string"('AWR Checked Date Vacancy -
',@CompName),"string"("old_awr"."AWRCheckDate"),"string"("new_awr"."AWRCheck
Date"))
    END IF
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."AWRvacancy"."AWRvacancy_update" IS
{CREATE TRIGGER AWRvacancy_update
after UPDATE OF AWRStatus, AWRCheckDate
ORDER 1 ON pears.AWRvacancy
REFERENCING OLD AS old_awr NEW AS new_awr
FOR each ROW
BEGIN
    DECLARE @CompName CHAR(250);
    SET @CompName=(SELECT string(Company.Name,' -
',vacancy.position,'(' ,vacancy.RefCode,' )') FROM Company KEY JOIN Employment
KEY JOIN vacancy WHERE vacancy.vacancyid = old_awr.vacancyID);
    -- audit
    IF UPDATE(AWRStatus) THEN
        CALL AuditLog('AWRVAC',old_awr.vacancyID,
        string('AWR Status Vacancy - ',@CompName),CASE old_awr.AWRStatus WHEN 1
THEN 'Not Known' WHEN 2 THEN 'AWR Applies' WHEN 3 THEN 'AWR not Applicable'
END,CASE new_awr.AWRStatus WHEN 1 THEN 'Not Known' WHEN 2 THEN 'AWR
```



```
Applies' WHEN 3 THEN 'AWR not Applicable'
    END);
END IF;
IF UPDATE(AWRCheckDate) THEN
    CALL AuditLog('AWRVAC',old_awr.vacancyID,
        string('AWR Checked Date Vacancy -
',@CompName),string(old_awr.AWRCheckDate),string(new_awr.AWRCheckDate));
    END IF
END
}
GO
```

From:

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

Permanent link:

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

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

