



pears.AWRcompany



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

Description

AWR settings for Company records. One-to-one relationship with Company table.

Columns

Column	Type	Null	Default	Comment
companyid	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		
LECode	char(50)	NULL		
AWRCheckDate	date	NULL		

Primary Key

- companyid

Foreign Keys

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

Referenced By

- No incoming foreign keys found.



Indexes

- No indexes found.

Triggers

Name	Timing	Event
AWRcompany_update	after	update of "AWRStatus", "AWRCheckDate", "LECode" order 1
AWRCompany_Insert	before	insert order 1
AWRCompany_UpdateTrim	before	update order 2

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."AWRcompany"
-- Table comment: AWR settings for Company records. One-to-one relationship
with Company table.
-- Statement count: 9

CREATE TABLE "pears"."AWRcompany" (
  "companyid"          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
, "LECode"             CHAR(50) NULL
, "AWRCheckDate"       DATE NULL
, PRIMARY KEY ("companyid" ASC)
)
GO

COMMENT ON TABLE "pears"."AWRcompany" IS
  'AWR settings for Company records. One-to-one relationship with Company
table.'
GO

ALTER TABLE "pears"."AWRcompany"
  ADD NOT NULL FOREIGN KEY "Company" ("companyid" ASC)
```



```
REFERENCES "pears"."Company" ("companyid")
ON DELETE CASCADE
GO

CREATE TRIGGER "AWRcompany_update" after UPDATE OF "AWRStatus",
"AWRCheckDate",
"LECode" ORDER 1 ON "pears"."AWRcompany"
REFERENCING OLD AS "old_awr" NEW AS "new_awr"
FOR each ROW
BEGIN
    DECLARE @CompName CHAR(250);
    SET @CompName = (SELECT "Name" FROM "company" WHERE "companyid" =
"old_awr"."companyID");
    -- audit
    IF UPDATE("AWRStatus") THEN
        CALL "AuditLog"('AWRCOMP',"old_awr"."companyID",
        "string"('AWR Status Company - ',@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("LECode") THEN
        CALL "AuditLog"('AWRCOMP',"old_awr"."companyID",
        "string"('AWR LE Code Company -
',@CompName),"old_awr"."LECode","new_awr"."LECode")
    END IF;
    IF UPDATE("AWRCheckDate") THEN
        CALL "AuditLog"('AWRCOMP',"old_awr"."companyID",
        "string"('AWR Checked Date Company -
',@CompName),"string"("old_awr"."AWRCheckDate"),"string"("new_awr"."AWRCheck
Date"))
    END IF
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."AWRcompany"."AWRcompany_update" IS
{CREATE TRIGGER AWRcompany_update
after UPDATE OF AWRStatus,
AWRCheckDate,
LECode ORDER 1 ON pears.AWRcompany
REFERENCING OLD AS old_awr NEW AS new_awr
FOR each ROW
BEGIN
    DECLARE @CompName CHAR(250);
```



```
SET @CompName=(SELECT Name FROM company WHERE companyid =
old_awr.companyID);
-- audit
IF UPDATE(AWRStatus) THEN
    CALL AuditLog('AWRCOMP',old_awr.companyID,
    string('AWR Status Company - ',@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(LECode) THEN
    CALL AuditLog('AWRCOMP',old_awr.companyID,
    string('AWR LE Code Company - ',@CompName),old_awr.LECode,new_awr.LECode)
END IF;
IF UPDATE(AWRCheckDate) THEN
    CALL AuditLog('AWRCOMP',old_awr.companyID,
    string('AWR Checked Date Company - ',@CompName),string(old_awr.AWRCheckDate),string(new_awr.AWRCheckDate))
END IF
END
}
GO

CREATE TRIGGER "AWRCompany_Insert" BEFORE INSERT ORDER 1 ON
"pears"."AWRCompany"
REFERENCING NEW AS "new_awr"
FOR each ROW
BEGIN
    DECLARE @CompName CHAR(250);
    SET @CompName = (SELECT "Name" FROM "company" WHERE "companyid" =
"new_awr"."companyID");
    SET "new_awr"."LECode" = "trim"("new_awr"."LECode");
    CALL "AuditLog"('AWRCOMP',"new_awr"."companyID",
    "string('AWR Status Company Insert - ',@CompName),'',CASE
"new_awr"."AWRStatus" WHEN 1 THEN 'Not Known' WHEN 2 THEN 'AWR Applies' WHEN
3 THEN 'AWR not Applicable' END);
    IF "isnull"("new_awr"."LECode",'') > '' THEN
        CALL "AuditLog"('AWRCOMP',"new_awr"."companyID",
        "string('AWR LE Code Company Insert - ',@CompName),'',"new_awr"."LECode")
    END IF;
    IF "new_awr"."AWRCheckDate" IS NOT NULL THEN
        CALL "AuditLog"('AWRCOMP',"new_awr"."companyID",
        "string('AWR Checked Date Company Insert - ',@CompName),'',"string("new_awr"."AWRCheckDate"))
```



```
END IF
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."AWRcompany"."AWRCompany_Insert" IS
{CREATE TRIGGER AWRCompany_Insert
  BEFORE INSERT ORDER 1 ON
pears.AWRCompany
REFERENCING NEW AS new_awr
FOR each ROW
BEGIN
  DECLARE @CompName CHAR(250);
  SET @CompName=(SELECT Name FROM company WHERE companyid =
new_awr.companyID);
  SET new_awr.LECode = TRIM(new_awr.LECode);
  CALL AuditLog('AWRCOMP',new_awr.companyID,
  string('AWR Status Company Insert - ',@CompName),'',CASE new_awr.AWRStatus
WHEN 1 THEN 'Not Known' WHEN 2 THEN 'AWR Applies' WHEN 3 THEN 'AWR not
Applicable'
  END);
  IF isnull(new_awr.LECode,'') > '' THEN
    CALL AuditLog('AWRCOMP',new_awr.companyID,
    string('AWR LE Code Company Insert - ',@CompName),'',new_awr.LECode)
  END IF;
  IF new_awr.AWRCheckDate IS NOT NULL THEN
    CALL AuditLog('AWRCOMP',new_awr.companyID,
    string('AWR Checked Date Company Insert - ',@CompName),'',string(new_awr.AWRCheckDate))
  END IF
END
}
GO

CREATE TRIGGER "AWRCompany_UpdateTrim" BEFORE UPDATE ORDER 2 ON
"pears"."AWRcompany"
REFERENCING NEW AS "new_awr"
FOR each ROW
BEGIN
  SET "new_awr"."LECode" = "trim"("new_awr"."LECode")
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."AWRcompany"."AWRCompany_UpdateTrim" IS
```



```
{CREATE TRIGGER AWRCompany_UpdateTrim
  BEFORE UPDATE ORDER 2 ON
pears.AWRCompany
REFERENCING NEW AS new_awr
FOR each ROW
BEGIN
  SET new_awr.LECode = TRIM(new_awr.LECode);
END
}
GO
```

From:

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

Permanent link:

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

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

