



pears.tag



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

Description

User defined field definitions (Questionnaires)

Columns

Column	Type	Null	Default	Comment
taglocation	char(3)	NOT NULL		
tagid	char(3)	NOT NULL		
tagtype	char(1)	NOT NULL		
description	char(60)	NOT NULL		
sortorder	smallint	NULL		
minstep	double	NULL		
units	char(10)	NULL		
required	smallint	NULL		
displaygroup	smallint	NULL	0	
fixed	smallint	NULL	0	
AuditFlag	smallint	NULL	0	
PublishToWeb	smallint	NULL	0	
PublicOnWeb	smallint	NULL	0	Viewable, but not editable, by 3rd parties on web
NonSearchable	smallint	NULL	0	
HeadingExpanded	smallint	NULL	0	
Expiry	smallint	NULL		
readonly	smallint	NULL		
LongDescription	long varchar	NULL		
HeadingAnswers	smallint	NULL	0	
Archived	tinyint	NULL		
ExpiryBehaviour	smallint	NULL		
BackgroundColour	integer	NULL	16777215	
FontColour	integer	NULL	0	
wizarddisplaygroup	smallint	NULL	0	
IncludeInKeywordSearch	tinyint	NOT NULL	0	



Column	Type	Null	Default	Comment
DependsOnTagid	char(3)	NULL		Ask if this tag is filled (in same taglocation), and DependsOnTagchoiceid is selected or null
DependsOnTagchoiceid	char(4)	NULL		Ask if this DependsOnTagid choice is selected or null
WebDescription	long varchar	NULL		
MinValue	double	NULL	0	
MaxValue	double	NULL	0	

Primary Key

- taglocation, tagid

Foreign Keys

- No outgoing foreign keys found.

Referenced By

Table	Constraint	Columns	Referenced columns
pears.criterion	tag	sourcelocation, tagid	taglocation, tagid
pears.DepartmentTag	TagKey	TagLocation, TagID	taglocation, tagid
pears.DocumentTag	TagKey	TagLocation, TagID	taglocation, tagid
pears.IQXNetTagGroupMember	tag	TagLocation, TagID	taglocation, tagid
pears.ReferenceMasterQuestion	tag	taglocation, tagid	taglocation, tagid
pears.tagchoice	tag	taglocation, tagid	taglocation, tagid
pears.TagCompliance	tag	taglocation, tagid	taglocation, tagid
pears.tagvalue	tag	taglocation, tagid	taglocation, tagid

Indexes

Name	Type	Columns	Detail
tag_sortorder	Index	taglocation, sortorder	

Triggers

Name	Timing	Event
WPKTagChange	after	insert,delete,update
TagInsertAudit	after	insert order 1
TagDeleteAudit	after	delete order 1



Name	Timing	Event
TagUpdateAudit	after	update order 1
TagKeyword	after	update of "IncludeInKeyWordSearch" order 20
Tag_UpdateDescriptionFormat	before	insert,update order 50

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."tag"
-- Table comment: User defined field definitions (Questionnaires)
-- Statement count: 17

CREATE TABLE "pears"."tag" (
    "taglocation" CHAR(3) NOT NULL
  ,"tagid" CHAR(3) NOT NULL
  ,"tagtype" CHAR(1) NOT NULL
  ,"description" CHAR(60) NOT NULL INLINE 35 PREFIX 8
  ,"sortorder" SMALLINT NULL
  ,"minstep" DOUBLE NULL
  ,"units" CHAR(10) NULL
  ,"required" SMALLINT NULL
  ,"displaygroup" SMALLINT NULL DEFAULT 0
  ,"fixed" SMALLINT NULL DEFAULT 0
  ,"AuditFlag" SMALLINT NULL DEFAULT 0
  ,"PublishToWeb" SMALLINT NULL DEFAULT 0
  ,"PublicOnWeb" SMALLINT NULL DEFAULT 0
  ,"NonSearchable" SMALLINT NULL DEFAULT 0
  ,"HeadingExpanded" SMALLINT NULL DEFAULT 0
  ,"Expiry" SMALLINT NULL
  ,"readonly" SMALLINT NULL
  ,"LongDescription" long VARCHAR NULL
  ,"HeadingAnswers" SMALLINT NULL DEFAULT 0
  ,"Archived" tinyint NULL
  ,"ExpiryBehaviour" SMALLINT NULL
  ,"BackgroundColour" INTEGER NULL DEFAULT 16777215
  ,"FontColour" INTEGER NULL DEFAULT 0
  ,"wizarddisplaygroup" SMALLINT NULL DEFAULT 0
  ,"IncludeInKeyWordSearch" tinyint NOT NULL DEFAULT 0
  ,"DependsOnTagid" CHAR(3) NULL
  ,"DependsOnTagchoiceid" CHAR(4) NULL
  ,"WebDescription" long VARCHAR NULL
  ,"MinValue" DOUBLE NULL DEFAULT 0
  ,"MaxValue" DOUBLE NULL DEFAULT 0
  ,PRIMARY KEY ("taglocation" ASC,"tagid" ASC)
)
```



GO

```
COMMENT ON COLUMN "pears"."tag"."PublicOnWeb" IS
    'Viewable, but not editable, by 3rd parties on web'
```

GO

```
COMMENT ON COLUMN "pears"."tag"."DependsOnTagid" IS
    'Ask if this tag is filled (in same taglocation), and
    DependsOnTagchoiceid is selected or null'
```

GO

```
COMMENT ON COLUMN "pears"."tag"."DependsOnTagchoiceid" IS
    'Ask if this DependsOnTagid choice is selected or null'
```

GO

```
COMMENT ON TABLE "pears"."tag" IS
    'User defined field definitions (Questionnaires)'
```

GO

```
CREATE INDEX "tag_sortorder" ON "pears"."tag"
    ( "taglocation","sortorder" )
```

GO

```
CREATE TRIGGER "WPKTagChange" after INSERT,DELETE,UPDATE ON
"pears"."Tag"
```

```
REFERENCING NEW AS "tg"
```

```
FOR each ROW
```

```
BEGIN
```

```
    CALL "WPKTrackChange"('Q',"tg"."taglocation")
```

```
END
```

GO

```
CREATE TRIGGER "TagInsertAudit" after INSERT ORDER 1 ON
"pears"."Tag"
```

```
REFERENCING NEW AS "new_tag"
```

```
FOR each ROW
```

```
BEGIN
```

```
    CALL "AuditLog"('TAG','', "string"('Question ', "new_tag"."description", '
Added id = ', "new_tag"."tagid", ' location =
', "new_tag"."taglocation"),'', 'Added')
```

```
END
```



GO

```
COMMENT TO PRESERVE FORMAT ON TRIGGER "pears"."tag"."TagInsertAudit" IS
{CREATE TRIGGER TagInsertAudit
  after INSERT ORDER 1 ON
Tag
REFERENCING NEW AS new_tag
FOR each ROW
BEGIN
  CALL AuditLog('TAG','',string('Question ',new_tag.description,' Added id =
',new_tag.tagid,' location = ', new_tag.taglocation),',' , 'Added')
END
}
GO
```

```
CREATE TRIGGER "TagDeleteAudit" after DELETE ORDER 1 ON
"pears"."tag"
REFERENCING OLD AS "old_tag"
FOR each ROW
BEGIN
  CALL "AuditLog"('TAG','',string('Question ', "old_tag"."description", '
Deleted id = ', "old_tag"."tagid", ' location = ', "old_tag"."taglocation"),
  "string"('sort order = ', "old_tag"."sortorder", ' type =
', "old_tag"."tagtype", ' Min step = ', "old_tag"."minstep", ' Units =
', "old_tag"."units", ' Required = ', "old_tag"."required",
  ' Group = ', "old_tag"."displaygroup", ' Audit = ', "old_tag"."auditflag", '
Web Publish = ', "old_tag"."PublishToweb", ' Web View =
', "old_tag"."publiconweb",
  ' Search Hide = ', "old_tag"."nonsearchable", ' Collapse =
', "old_tag"."HeadingExpanded", ' Expiry Lead = ', "old_tag"."expiry", ' Long
Description = ', "old_tag"."longdescription"),',' )
END
GO
```

```
COMMENT TO PRESERVE FORMAT ON TRIGGER "pears"."tag"."TagDeleteAudit" IS
{CREATE TRIGGER TagDeleteAudit
  after DELETE ORDER 1 ON
tag
REFERENCING OLD AS old_tag
FOR each ROW
BEGIN
  CALL AuditLog('TAG','',string('Question ',old_tag.description,' Deleted id
= ',old_tag.tagid,' location = ', old_tag.taglocation),
string('sort order = ',old_tag.sortorder,' type = ',old_tag.tagtype,' Min
step = ',old_tag.minstep,' Units = ',old_tag.units,' Required =
```



```
',old_tag.required,  
' Group = ',old_tag.displaygroup,' Audit = ',old_tag.auditflag,' Web Publish  
= ',old_tag.PublishToweb,' Web View = ',old_tag.publiconweb,  
' Search Hide = ',old_tag.nonsearchable,' Collapse =  
' ,old_tag.HeadingExpanded,' Expiry Lead = ',old_tag.expiry,' Long  
Description = ',old_tag.longdescription ),')  
END  
}  
GO
```

```
CREATE TRIGGER "TagUpdateAudit" after UPDATE ORDER 1 ON  
"pears"."tag"  
REFERENCING OLD AS "old_tag" NEW AS "new_tag"  
FOR each ROW  
BEGIN  
    CALL "AuditLog"('TAG','',"string"('Question ','"new_tag"."description",'  
Changed id = ','"old_tag"."tagid",' location = ','"old_tag"."taglocation"),  
    "string"('description = ','"old_tag"."description",' sort order =  
' ,'"old_tag"."sortorder",' type = ','"old_tag"."tagtype",' Min step =  
' ,'"old_tag"."minstep",' Units = ','"old_tag"."units",' Required =  
' ,'"old_tag"."required",  
    ' Group = ','"old_tag"."displaygroup",' Audit = ','"old_tag"."auditflag",'  
Web Publish = ','"old_tag"."PublishToweb",' Web View =  
' ,'"old_tag"."publiconweb",  
    ' Search Hide = ','"old_tag"."nonsearchable",' Collapse =  
' ,'"old_tag"."HeadingExpanded",' Expiry Lead = ','"old_tag"."expiry",' Long  
Description = ','"old_tag"."longdescription",' Web Description =  
' ,'"old_tag"."webdescription"),  
    "string"('description = ','"new_tag"."description",' sort order =  
' ,'"new_tag"."sortorder",' type = ','"new_tag"."tagtype",' Min step =  
' ,'"new_tag"."minstep",' Units = ','"new_tag"."units",' Required =  
' ,'"new_tag"."required",  
    ' Group = ','"new_tag"."displaygroup",' Audit = ','"new_tag"."auditflag",'  
Web Publish = ','"new_tag"."PublishToweb",' Web View =  
' ,'"new_tag"."publiconweb",  
    ' Search Hide = ','"new_tag"."nonsearchable",' Collapse =  
' ,'"new_tag"."HeadingExpanded",' Expiry Lead = ','"new_tag"."expiry",' Long  
Description = ','"new_tag"."longdescription",' Web Description =  
' ,'"new_tag"."webdescription"))  
END  
GO
```

```
COMMENT TO PRESERVE FORMAT ON TRIGGER "pears"."tag"."TagUpdateAudit" IS  
{CREATE TRIGGER TagUpdateAudit  
after UPDATE ORDER 1 ON  
tag
```



```
REFERENCING OLD AS old_tag NEW AS new_tag
FOR each ROW
BEGIN
    CALL AuditLog('TAG','',string('Question ',new_tag.description,' Changed id
= ',old_tag.tagid,' location = ', old_tag.taglocation),
string('description = ',old_tag.description,' sort order =
',old_tag.sortorder,' type = ',old_tag.tagtype,' Min step =
',old_tag.minstep,' Units = ',old_tag.units,' Required = ',old_tag.required,
' Group = ',old_tag.displaygroup,' Audit = ',old_tag.auditflag,' Web Publish
= ',old_tag.PublishToweb,' Web View = ',old_tag.publiconweb,
' Search Hide = ',old_tag.nonsearchable,' Collapse =
',old_tag.HeadingExpanded,' Expiry Lead = ',old_tag.expiry,' Long
Description = ',old_tag.longdescription ,' Web Description =
',old_tag.webdescription),
string('description = ',new_tag.description,' sort order =
',new_tag.sortorder,' type = ',new_tag.tagtype,' Min step =
',new_tag.minstep,' Units = ',new_tag.units,' Required = ',new_tag.required,
' Group = ',new_tag.displaygroup,' Audit = ',new_tag.auditflag,' Web Publish
= ',new_tag.PublishToweb,' Web View = ',new_tag.publiconweb,
' Search Hide = ',new_tag.nonsearchable,' Collapse =
',new_tag.HeadingExpanded,' Expiry Lead = ',new_tag.expiry,' Long
Description = ',new_tag.longdescription,' Web Description =
',new_tag.webdescription ))
END
}
GO
```

```
CREATE TRIGGER "TagKeyword" after UPDATE OF "IncludeInKeyWordSearch"
ORDER 20 ON
"pears"."Tag"
REFERENCING NEW AS "NewRow"
FOR each ROW
BEGIN
    CASE WHEN("NewRow"."TagLocation" IN( 'P' ) OR "NewRow"."TagLocation" LIKE
'A%') THEN
        CASE
            WHEN "NewRow"."TagType" <> 'M'
                OR(SELECT "count"() FROM "TagChoice" AS "tc" WHERE "tc"."TagLocation" =
"NewRow"."TagLocation" AND "tc"."TagID" = "NewRow"."TagID" AND
"tc"."TagChoiceID" = '_') = 0 THEN // no DEFAULT TagChoice so ONLY People
WITH TagValue records need updating
                UPDATE "PersonKeyWords" AS "pkw" SET "TagRefreshRequired" = 1 FROM
"PersonKeyWords" AS "pkw" JOIN "TagValue" AS "tv" ON "tv"."TagLocation" =
"NewRow"."TagLocation" AND "tv"."TagID" = "NewRow"."TagID" AND "tv"."ID" =
"pkw"."PersonID"
            WHEN "NewRow"."TagLocation" LIKE 'A%' THEN // ONLY need TO SET VALUES
FOR that department
```



```
UPDATE "PersonKeyWords" AS "pkw" SET "TagRefreshRequired" = 1 FROM
"PersonKeyWords" AS "pkw" JOIN "Search" AS "s" ON "s"."PersonID" =
"pkw"."PersonID" AND "s"."DepartmentID" =
"substring"("NewRow"."TagLocation",2)
  WHEN "NewRow"."TagLocation" = 'P' THEN // need TO SET VALUES FOR ALL
    UPDATE "PersonKeyWords" AS "pkw" SET "TagRefreshRequired" = 1
  END CASE
WHEN("NewRow"."TagLocation" LIKE 'C%') THEN
  CASE
    WHEN "NewRow"."TagType" <> 'M'
      OR(SELECT "count"() FROM "TagChoice" AS "tc" WHERE "tc"."TagLocation" =
"NewRow"."TagLocation" AND "tc"."TagID" = "NewRow"."TagID" AND
"tc"."TagChoiceID" = '_' ) = 0 THEN // no DEFAULT TagChoice so ONLY Company
WITH TagValue records need updating
        UPDATE "CompanyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1 FROM
"CompanyKeyWords" AS "pkw" JOIN "TagValue" AS "tv" ON "tv"."TagLocation" =
"NewRow"."TagLocation" AND "tv"."TagID" = "NewRow"."TagID" AND "tv"."ID" =
"pkw"."CompanyID"
        WHEN "length"("NewRow"."TagLocation") > 1 THEN // ONLY need TO SET
VALUES FOR that department
          UPDATE "CompanyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1 FROM
"CompanyKeyWords" AS "pkw" JOIN "company" AS "c" ON("c"."company" =
"pkw"."CompanyID"), "company" AS "c" KEY JOIN "staff" AS "s" WHERE
"s"."Defaultdepartid" = "substring"("NewRow"."TagLocation",2,2)
        WHEN "NewRow"."TagLocation" = 'C' THEN // need TO SET VALUES FOR ALL
          UPDATE "CompanyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1
        END CASE
    WHEN("NewRow"."TagLocation" LIKE 'V%') THEN
      CASE
        WHEN "NewRow"."TagType" <> 'M'
          OR(SELECT "count"() FROM "TagChoice" AS "tc" WHERE "tc"."TagLocation" =
"NewRow"."TagLocation" AND "tc"."TagID" = "NewRow"."TagID" AND
"tc"."TagChoiceID" = '_' ) = 0 THEN // no DEFAULT TagChoice so ONLY vacancy
WITH TagValue records need updating
            UPDATE "VacancyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1 FROM
"VacancyKeyWords" AS "pkw" JOIN "TagValue" AS "tv" ON "tv"."TagLocation" =
"NewRow"."TagLocation" AND "tv"."TagID" = "NewRow"."TagID" AND "tv"."ID" =
"pkw"."VacancyID"
            WHEN "length"("NewRow"."TagLocation") > 1 THEN // ONLY need TO SET
VALUES FOR that department
              UPDATE "VacancyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1 FROM
"VacancyKeyWords" AS "pkw" JOIN "vacancy" AS "s" ON "s"."vacancyID" =
"pkw"."vacancyID" AND "s"."DepartmentID" =
"substring"("NewRow"."TagLocation",2,2)
            WHEN "NewRow"."TagLocation" = 'V' THEN // need TO SET VALUES FOR ALL
              UPDATE "VacancyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1
            END CASE
          END CASE
```



```
END
GO
```

```
COMMENT TO PRESERVE FORMAT ON TRIGGER "pears"."tag"."TagKeyword" IS
{CREATE TRIGGER TagKeyword
  AFTER UPDATE OF IncludeInKeyWordSearch ORDER 20 ON
  pears.Tag
  REFERENCING NEW AS NewRow
  FOR EACH ROW
  BEGIN
    CASE WHEN (NewRow.TagLocation IN ('P') OR NewRow.TagLocation LIKE 'A%' )
  THEN
    CASE
      WHEN NewRow.TagType != 'M' OR
        (SELECT COUNT(*) FROM TagChoice tc WHERE tc.TagLocation =
NewRow.TagLocation AND tc.TagID = NewRow.TagID AND tc.TagChoiceID = '_' ) =
0 THEN // no DEFAULT TagChoice so ONLY People WITH TagValue records need
updating
        UPDATE PersonKeyWords pkw SET TagRefreshRequired = 1 FROM
PersonKeyWords pkw JOIN TagValue tv ON tv.TagLocation = NewRow.TagLocation
AND tv.TagID = NewRow.TagID AND tv.ID = pkw.PersonID
        WHEN NewRow.TagLocation LIKE 'A%' THEN // ONLY need TO SET VALUES
FOR that department
          UPDATE PersonKeyWords pkw SET TagRefreshRequired = 1 FROM
PersonKeyWords pkw JOIN SEARCH s ON s.PersonID = pkw.PersonID AND
s.DepartmentID = SUBSTRING(NewRow.TagLocation,2)
        WHEN NewRow.TagLocation = 'P' THEN // need TO SET VALUES FOR ALL
          UPDATE PersonKeyWords pkw SET TagRefreshRequired = 1
        END CASE;
    WHEN (NewRow.TagLocation LIKE 'C%' ) THEN
      CASE
        WHEN NewRow.TagType != 'M' OR
          (SELECT COUNT(*) FROM TagChoice tc WHERE tc.TagLocation =
NewRow.TagLocation AND tc.TagID = NewRow.TagID AND tc.TagChoiceID = '_' ) =
0 THEN // no DEFAULT TagChoice so ONLY Company WITH TagValue records need
updating
          UPDATE CompanyKeyWords pkw SET TagRefreshRequired = 1 FROM
CompanyKeyWords pkw JOIN TagValue tv ON tv.TagLocation = NewRow.TagLocation
AND tv.TagID = NewRow.TagID AND tv.ID = pkw.CompanyID
          WHEN LENGTH(NewRow.TagLocation)>1 THEN // ONLY need TO SET VALUES
FOR that department
            UPDATE CompanyKeyWords pkw SET TagRefreshRequired = 1 FROM
CompanyKeyWords pkw JOIN company c ON (c.company = pkw.CompanyID) , company
c KEY JOIN staff s WHERE s.Defaultdepartid =
SUBSTRING(NewRow.TagLocation,2,2)
            WHEN NewRow.TagLocation = 'C' THEN // need TO SET VALUES FOR ALL
              UPDATE CompanyKeyWords pkw SET TagRefreshRequired = 1
```



```
        END CASE;
    WHEN (NewRow.TagLocation LIKE 'V%' ) THEN
        CASE
            WHEN NewRow.TagType != 'M' OR
                (SELECT COUNT(*) FROM TagChoice tc WHERE tc.TagLocation =
NewRow.TagLocation AND tc.TagID = NewRow.TagID AND tc.TagChoiceID = '_' ) =
0 THEN // no DEFAULT TagChoice so ONLY vacancy WITH TagValue records need
updating
                UPDATE VacancyKeyWords pkw SET TagRefreshRequired = 1 FROM
VacancyKeyWords pkw JOIN TagValue tv ON tv.TagLocation = NewRow.TagLocation
AND tv.TagID = NewRow.TagID AND tv.ID = pkw.VacancyID
                WHEN LENGTH(NewRow.TagLocation)>1 THEN // ONLY need TO SET VALUES
FOR that department
                        UPDATE VacancyKeyWords pkw SET TagRefreshRequired = 1 FROM
VacancyKeyWords pkw JOIN vacancy s ON s.vacancyID = pkw.vacancyID AND
s.DepartmentID = SUBSTRING(NewRow.TagLocation,2,2)
                        WHEN NewRow.TagLocation = 'V' THEN // need TO SET VALUES FOR ALL
                                UPDATE VacancyKeyWords pkw SET TagRefreshRequired = 1
        END CASE;
    END CASE;
END
}
GO

CREATE TRIGGER "Tag_UpdateDescriptionFormat" BEFORE INSERT,UPDATE ORDER 50
ON
"pears"."Tag"
REFERENCING NEW AS "new_tag"
FOR each ROW
BEGIN
    SET "new_tag"."description" = "trim"("new_tag"."description")
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."tag"."Tag_UpdateDescriptionFormat" IS
{CREATE TRIGGER Tag_UpdateDescriptionFormat
BEFORE INSERT, UPDATE ORDER 50 ON
pears.Tag
REFERENCING NEW AS new_tag
FOR each ROW
BEGIN
    SET new_tag.description = TRIM(new_tag.description);
END
}
```



G0

From:

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

Permanent link:

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

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

