



pears.tagchoice



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

Description

Choices for picklist responses to User Defined fields

Columns

Column	Type	Null	Default	Comment
taglocation	char(3)	NOT NULL		
tagid	char(3)	NOT NULL		
tagchoiceid	char(4)	NOT NULL		
description	char(60)	NOT NULL		
sortorder	smallint	NULL		
subchoice	smallint	NULL	0	
value	double	NULL		
aliases	char(250)	NULL		
fixed	smallint	NULL	0	
DoNotPublishToWeb	smallint	NULL	0	
analysis	char(20)	NULL		
tagchoiceparentid	char(4)	NULL		
LongDescription	long varchar	NULL		
readonly	smallint	NULL		

Primary Key

- taglocation, tagid, tagchoiceid

Foreign Keys

Constraint	Columns	References	Delete/update action
tag	taglocation, tagid	pears.tag (taglocation, tagid)	NOT NULL; ON DELETE CASCADE



Constraint	Columns	References	Delete/update action
tagchoice_parent	taglocation, tagid, tagchoiceparentid	pears.tagchoice (taglocation, tagid, tagchoiceid)	ON DELETE CASCADE

Referenced By

Table	Constraint	Columns	Referenced columns
pears.criterion	tagchoice	sourcelocation, tagid, tagchoiceid	taglocation, tagid, tagchoiceid
pears.tagchoice	tagchoice_parent	taglocation, tagid, tagchoiceparentid	taglocation, tagid, tagchoiceid
pears.tagvalue	tagchoice	taglocation, tagid, tagchoiceid	taglocation, tagid, tagchoiceid

Indexes

Name	Type	Columns	Detail
tagchoice_sortorder	Index	taglocation, tagid, sortorder	
tagchoice_description	Index	taglocation, description	

Triggers

Name	Timing	Event
WPKTagChoiceChange	after	insert,delete,update
TagChoiceInsertAudit	after	insert order 1
TagChoiceDeleteAudit	after	delete order 1
TagChoiceUpdateAudit	after	update order 1
TagChoiceKeyword	after	insert,delete,update order 20

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."tagchoice"
-- Table comment: Choices for picklist responses to User Defined fields
-- Statement count: 15
```

```
CREATE TABLE "pears"."tagchoice" (
    "taglocation"          CHAR(3) NOT NULL
  , "tagid"                CHAR(3) NOT NULL
  , "tagchoiceid"          CHAR(4) NOT NULL
  , "description"          CHAR(60) NOT NULL
  , "sortorder"            SMALLINT NULL
  , "subchoice"            SMALLINT NULL DEFAULT 0
  , "value"                DOUBLE NULL
  , "aliases"              CHAR(250) NULL
```



```
, "fixed"                SMALLINT NULL DEFAULT 0
, "DoNotPublishToWeb"    SMALLINT NULL DEFAULT 0
, "analysis"             CHAR(20) NULL
, "tagchoiceparentid"    CHAR(4) NULL
, "LongDescription"      long VARCHAR NULL
, "readonly"             SMALLINT NULL
, PRIMARY KEY ("taglocation" ASC, "tagid" ASC, "tagchoiceid" ASC)
)
GO

COMMENT ON TABLE "pears"."tagchoice" IS
    'Choices for picklist responses to User Defined fields'
GO

ALTER TABLE "pears"."tagchoice"
    ADD NOT NULL FOREIGN KEY "tag" ("taglocation" ASC, "tagid" ASC)
    REFERENCES "pears"."tag" ("taglocation", "tagid")
    ON DELETE CASCADE
GO

ALTER TABLE "pears"."tagchoice"
    ADD FOREIGN KEY "tagchoice_parent" ("taglocation" ASC, "tagid"
ASC, "tagchoiceparentid" ASC)
    REFERENCES "pears"."tagchoice" ("taglocation", "tagid", "tagchoiceid")
    ON DELETE CASCADE
GO

CREATE INDEX "tagchoice_sortorder" ON "pears"."tagchoice"
    ( "taglocation", "tagid", "sortorder" )
GO

CREATE INDEX "tagchoice_description" ON "pears"."tagchoice"
    ( "taglocation", "description" )
GO

CREATE TRIGGER "WPKTagChoiceChange" after INSERT,DELETE,UPDATE ON
"pears"."TagChoice"
REFERENCING NEW AS "tc"
FOR each ROW
BEGIN
    CALL "WPKTrackChange"('Q', "tc"."taglocation")
END
```



GO

```
CREATE TRIGGER "TagChoiceInsertAudit" after INSERT ORDER 1 ON
"pears"."TagChoice"
REFERENCING NEW AS "new_tag"
FOR each ROW
BEGIN
    CALL "AuditLog"('TAGCHOICE','',string('Question Choice
',"new_tag"."description",' Added id = ', "new_tag"."tagid",' location =
',"new_tag"."taglocation",' choice id =
',"new_tag"."tagchoiceid"),'', 'Added')
END
GO
```

```
COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."tagchoice"."TagChoiceInsertAudit" IS
{CREATE TRIGGER TagChoiceInsertAudit
after INSERT ORDER 1 ON
TagChoice
REFERENCING NEW AS new_tag
FOR each ROW
BEGIN
    CALL AuditLog('TAGCHOICE','',string('Question Choice
',new_tag.description,' Added id = ',new_tag.tagid,' location = ',
new_tag.taglocation,' choice id = ',new_tag.tagchoiceid),'', 'Added')
END
}
GO
```

```
CREATE TRIGGER "TagChoiceDeleteAudit" after DELETE ORDER 1 ON
"pears"."tagchoice"
REFERENCING OLD AS "old_tag"
FOR each ROW
BEGIN
    CALL "AuditLog"('TAGCHOICE','',string('Question Choice
',"old_tag"."description",' Deleted id = ', "old_tag"."tagid",' location =
',"old_tag"."taglocation",
' choice id = ', "old_tag"."tagchoiceid"),string('Autofill =
',"old_tag"."aliases",' value = ', "old_tag"."value",' Web Hide =
',"old_tag"."DoNotPublishToWeb",' analysis = ', "old_tag"."analysis",'
subchoice = ', "old_tag"."subchoice"),'')
END
GO
```



```
COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."tagchoice"."TagChoiceDeleteAudit" IS
{CREATE TRIGGER TagChoiceDeleteAudit
  after DELETE ORDER 1 ON
tagchoice
REFERENCING OLD AS old_tag
FOR each ROW
BEGIN
  CALL AuditLog('TAGCHOICE','',string('Question Choice
',old_tag.description,' Deleted id = ',old_tag.tagid,' location = ',
old_tag.taglocation,
' choice id = ',old_tag.tagchoiceid),string('Autofill = ',old_tag.aliases,'
value = ',old_tag.value,' Web Hide = ',old_tag.DoNotPublishToWeb,' analysis
= ',old_tag.analysis,' subchoice = ',old_tag.subchoice),'')
END
}
GO
```

```
CREATE TRIGGER "TagChoiceUpdateAudit" after UPDATE ORDER 1 ON
"pears"."tagchoice"
REFERENCING OLD AS "old_tag" NEW AS "new_tag"
FOR each ROW
BEGIN
  CALL "AuditLog"('TAGCHOICE','',string('Question Choice
',"new_tag"."description",' Changed id = ',old_tag.tagid,' location =
',"old_tag"."taglocation",' choice id = ',old_tag.tagchoiceid),
  "string"('Description = ',old_tag.description,' Autofill =
',"old_tag"."aliases",' value = ',old_tag.value,' Web Hide =
',"old_tag"."DoNotPublishToWeb",' analysis = ',old_tag.analysis,'
subchoice = ',old_tag.subchoice,' readonly = ',old_tag.readonly)),
  "string"('Description = ',new_tag.description,' Autofill =
',"new_tag"."aliases",' value = ',new_tag.value,' Web Hide =
',"new_tag"."DoNotPublishToWeb",' analysis = ',new_tag.analysis,'
subchoice = ',new_tag.subchoice,' readonly = ',new_tag.readonly)))
END
GO
```

```
COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."tagchoice"."TagChoiceUpdateAudit" IS
{CREATE TRIGGER TagChoiceUpdateAudit
  after UPDATE ORDER 1 ON
tagchoice
REFERENCING OLD AS old_tag NEW AS new_tag
FOR each ROW
BEGIN
  CALL AuditLog('TAGCHOICE','',string('Question Choice
```



```
',new_tag.description,' Changed id = ',old_tag.tagid,' location = ',  
old_tag.taglocation,' choice id = ',old_tag.tagchoiceid)  
,string('Description = ',old_tag.description,' Autofill =  
,old_tag.aliases,' value = ',old_tag.value,' Web Hide =  
,old_tag.DoNotPublishToWeb,' analysis = ',old_tag.analysis,' subchoice =  
,old_tag.subchoice,' readonly = ',old_tag.readonly),  
string('Description = ',new_tag.description,' Autofill = ',new_tag.aliases,'  
value = ',new_tag.value,' Web Hide = ',new_tag.DoNotPublishToWeb,' analysis  
= ',new_tag.analysis,' subchoice = ',new_tag.subchoice,' readonly =  
,new_tag.readonly))  
END  
}  
GO
```

```
CREATE TRIGGER "TagChoiceKeyword" after INSERT,DELETE,UPDATE ORDER 20 ON  
"pears"."TagChoice"  
REFERENCING OLD AS "OldRow" NEW AS "NewRow"  
FOR each ROW  
BEGIN  
    CASE WHEN deleting THEN  
        CASE WHEN("OldRow"."TagLocation" = 'P' OR "OldRow"."TagLocation" LIKE  
'A%') AND(SELECT "IncludeInKeyWordSearch" FROM "Tag" WHERE  
"Tag"."TagLocation" = "OldRow"."TagLocation" AND "Tag"."TagID" =  
"OldRow"."TagID") = 1 THEN  
            CASE  
                WHEN "OldRow"."TagChoiceID" <> '_' THEN // NOT DEFAULT TagChoice so  
ONLY People WITH related TagValue records need updating  
                UPDATE "PersonKeyWords" AS "pkw" SET "TagRefreshRequired" = 1 FROM  
"PersonKeyWords" AS "pkw" JOIN "TagValue" AS "tv" ON "tv"."TagLocation" =  
"OldRow"."TagLocation" AND "tv"."TagID" = "OldRow"."TagID" AND  
"tv"."TagChoiceID" = "OldRow"."TagChoiceID" AND "tv"."ID" = "pkw"."PersonID"  
                WHEN "OldRow"."TagLocation" LIKE 'A%' THEN // ONLY need TO SET VALUES  
FOR those registered WITH 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"("OldRow"."TagLocation",2)  
                WHEN "NewRow"."TagLocation" = 'P' THEN // need TO SET VALUES FOR ALL  
                UPDATE "PersonKeyWords" AS "pkw" SET "TagRefreshRequired" = 1  
            END CASE  
        WHEN("OldRow"."TagLocation" LIKE 'C%') AND(SELECT  
"IncludeInKeyWordSearch" FROM "Tag" WHERE "Tag"."TagLocation" =  
"OldRow"."TagLocation" AND "Tag"."TagID" = "OldRow"."TagID") = 1 THEN  
            CASE  
                WHEN "OldRow"."TagChoiceID" <> '_' THEN // NOT DEFAULT TagChoice so  
ONLY People WITH related TagValue records need updating  
                UPDATE "CompanyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1 FROM
```



```
"CompanyKeyWords" AS "pkw" JOIN "TagValue" AS "tv" ON "tv"."TagLocation" =
"OldRow"."TagLocation" AND "tv"."TagID" = "OldRow"."TagID" AND
"tv"."TagChoiceID" = "OldRow"."TagChoiceID" AND "tv"."ID" =
"pkw"."CompanyID"
    WHEN "length"("OldRow"."TagLocation") > 1 THEN // ONLY need TO SET
VALUES FOR those registered WITH that department
        UPDATE "CompanyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1 FROM
"CompanyKeyWords" AS "pkw" JOIN "Company" AS "c" ON("c"."CompanyID" =
"pkw"."CompanyID"), "Company" AS "c" KEY JOIN "staff" AS "s" WHERE
"s"."Defaultdepartid" = "substring"("OldRow"."TagLocation",2)
    WHEN "NewRow"."TagLocation" = 'C' THEN // need TO SET VALUES FOR ALL
        UPDATE "CompanyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1
    END CASE
    WHEN("OldRow"."TagLocation" LIKE 'V%') AND(SELECT
"IncludeInKeywordSearch" FROM "Tag" WHERE "Tag"."TagLocation" =
"OldRow"."TagLocation" AND "Tag"."TagID" = "OldRow"."TagID") = 1 THEN
    CASE
        WHEN "OldRow"."TagChoiceID" <> '_' THEN // NOT DEFAULT TagChoice so
ONLY People WITH related TagValue records need updating
            UPDATE "VacancyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1 FROM
"VacancyKeyWords" AS "pkw" JOIN "TagValue" AS "tv" ON "tv"."TagLocation" =
"OldRow"."TagLocation" AND "tv"."TagID" = "OldRow"."TagID" AND
"tv"."TagChoiceID" = "OldRow"."TagChoiceID" AND "tv"."ID" =
"pkw"."VacancyID"
            WHEN "length"("OldRow"."TagLocation") > 1 THEN // ONLY need TO SET
VALUES FOR those registered WITH 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"("OldRow"."TagLocation",2)
            WHEN "NewRow"."TagLocation" = 'P' THEN // need TO SET VALUES FOR ALL
                UPDATE "VacancyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1
            END CASE
        END CASE // WHEN ..... - FOR other tag locations
    WHEN inserting THEN
        CASE WHEN("NewRow"."TagLocation" = 'P' OR "NewRow"."TagLocation" LIKE
'A%') AND(SELECT "IncludeInKeywordSearch" FROM "Tag" WHERE
"Tag"."TagLocation" = "NewRow"."TagLocation" AND "Tag"."TagID" =
"NewRow"."TagID") = 1 THEN
            IF "NewRow"."TagChoiceID" = '_' THEN // we are ONLY concerned WITH NEW
DEFAULT choices - anything ELSE will be dealt WITH BY the TagValue TRIGGER
                IF "NewRow"."TagLocation" LIKE 'A%' THEN // ONLY need TO SET VALUES
FOR those registered WITH 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)
                ELSE // everyone will need the TagKeyWords re-evaluated
```



```
        UPDATE "PersonKeyWords" AS "pkw" SET "TagRefreshRequired" = 1
    END IF
END IF
WHEN("NewRow"."TagLocation" LIKE 'C%') AND(SELECT
"IncludeInKeyWordSearch" FROM "Tag" WHERE "Tag"."TagLocation" =
"NewRow"."TagLocation" AND "Tag"."TagID" = "NewRow"."TagID") = 1 THEN
    IF "NewRow"."TagChoiceID" = '_' THEN // we are ONLY concerned WITH NEW
DEFAULT choices - anything ELSE will be dealt WITH BY the TagValue TRIGGER
        IF "length"("NewRow"."TagLocation") > 1 THEN // ONLY need TO SET
VALUES FOR those registered WITH that department
            UPDATE "CompanyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1
FROM "CompanyKeyWords" AS "pkw" JOIN "Company" AS "c" ON("c"."CompanyID" =
"pkw"."CompanyID"), "company" AS "c" KEY JOIN "staff" AS "s" WHERE
"s"."Defaultdepartid" = "substring"("NewRow"."TagLocation",2)
        ELSE // everyone will need the TagKeyWords re-evaluated
            UPDATE "CompanyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1
        END IF
    END IF
    WHEN("NewRow"."TagLocation" LIKE 'V%') AND(SELECT
"IncludeInKeyWordSearch" FROM "Tag" WHERE "Tag"."TagLocation" =
"NewRow"."TagLocation" AND "Tag"."TagID" = "NewRow"."TagID") = 1 THEN
        IF "NewRow"."TagChoiceID" = '_' THEN // we are ONLY concerned WITH NEW
DEFAULT choices - anything ELSE will be dealt WITH BY the TagValue TRIGGER
            IF "length"("NewRow"."TagLocation") > 1 THEN // ONLY need TO SET
VALUES FOR those registered WITH 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)
            ELSE // everyone will need the TagKeyWords re-evaluated
                UPDATE "VacancyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1
            END IF
        END IF
    END CASE
    WHEN updating THEN
        IF "NewRow"."Description" <> "OldRow"."Description" // we are ONLY
concerned IF the Description has changed
            AND("NewRow"."TagLocation" = 'P' OR "NewRow"."TagLocation" LIKE 'A%')
            AND(SELECT "IncludeInKeyWordSearch" FROM "Tag" WHERE
"Tag"."TagLocation" = "NewRow"."TagLocation" AND "Tag"."TagID" =
"NewRow"."TagID") = 1 THEN
                CASE
                    WHEN "OldRow"."TagChoiceID" <> '_' THEN // NOT DEFAULT TagChoice so
ONLY People WITH related TagValue records need updating
                        UPDATE "PersonKeyWords" AS "pkw" SET "TagRefreshRequired" = 1 FROM
"PersonKeyWords" AS "pkw" JOIN "TagValue" AS "tv" ON "tv"."TagLocation" =
"OldRow"."TagLocation" AND "tv"."TagID" = "OldRow"."TagID" AND
"tv"."TagChoiceID" = "OldRow"."TagChoiceID" AND "tv"."ID" = "pkw"."PersonID"
```



```
    WHEN "NewRow"."TagLocation" LIKE 'A%' THEN // we ONLY need TO deal
WITH the 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
    ELSE IF "NewRow"."Description" <> "OldRow"."Description" // we are ONLY
concerned IF the Description has changed
    AND("NewRow"."TagLocation" LIKE 'C%')
    AND(SELECT "IncludeInKeywordSearch" FROM "Tag" WHERE
"Tag"."TagLocation" = "NewRow"."TagLocation" AND "Tag"."TagID" =
"NewRow"."TagID") = 1 THEN
    CASE
    WHEN "OldRow"."TagChoiceID" <> '_' THEN // NOT DEFAULT TagChoice so
ONLY People WITH related TagValue records need updating
    UPDATE "CompanyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1
FROM "CompanyKeyWords" AS "pkw" JOIN "TagValue" AS "tv" ON
"tv"."TagLocation" = "OldRow"."TagLocation" AND "tv"."TagID" =
"OldRow"."TagID" AND "tv"."TagChoiceID" = "OldRow"."TagChoiceID" AND
"tv"."ID" = "pkw"."CompanyID"
    WHEN "NewRow"."TagLocation" LIKE 'A%' THEN // we ONLY need TO deal
WITH the department
    UPDATE "CompanyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1
FROM "CompanyKeyWords" AS "pkw" JOIN "Company" AS "s" ON("s"."CompanyID" =
"pkw"."CompanyID"), "Company" AS "s" KEY JOIN "Staff" AS "s" WHERE
"s"."Defaultdepartid" = "substring"("NewRow"."TagLocation",2)
    WHEN "NewRow"."TagLocation" = 'C' THEN // need TO SET VALUES FOR ALL
    UPDATE "CompanyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1
    END CASE
    ELSE IF "NewRow"."Description" <> "OldRow"."Description" // we are
ONLY concerned IF the Description has changed
    AND("NewRow"."TagLocation" LIKE 'V%')
    AND(SELECT "IncludeInKeywordSearch" FROM "Tag" WHERE
"Tag"."TagLocation" = "NewRow"."TagLocation" AND "Tag"."TagID" =
"NewRow"."TagID") = 1 THEN
    CASE
    WHEN "OldRow"."TagChoiceID" <> '_' THEN // NOT DEFAULT TagChoice
so ONLY People WITH related TagValue records need updating
    UPDATE "VacancyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1
FROM "VacancyKeyWords" AS "pkw" JOIN "TagValue" AS "tv" ON
"tv"."TagLocation" = "OldRow"."TagLocation" AND "tv"."TagID" =
"OldRow"."TagID" AND "tv"."TagChoiceID" = "OldRow"."TagChoiceID" AND
"tv"."ID" = "pkw"."Vacancyid"
    WHEN "length"("NewRow"."TagLocation") > 1 THEN // we ONLY need TO
deal WITH the 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)
        WHEN "NewRow"."TagLocation" = 'V' THEN // need TO SET VALUES FOR
ALL
        UPDATE "VacancyKeyWords" AS "pkw" SET "TagRefreshRequired" = 1
        END CASE
        END IF END IF END IF
    END CASE
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER "pears"."tagchoice"."TagChoiceKeyword"
IS
{CREATE TRIGGER TagChoiceKeyword
AFTER INSERT, UPDATE, DELETE ORDER 20 ON
pears.TagChoice
REFERENCING OLD AS OldRow NEW AS NewRow
FOR EACH ROW
BEGIN
    CASE WHEN DELETING THEN
        CASE WHEN (OldRow.TagLocation = 'P' OR OldRow.TagLocation LIKE 'A%' )
AND (SELECT IncludeInKeywordSearch FROM Tag WHERE Tag.TagLocation =
OldRow.TagLocation AND Tag.TagID = OldRow.TagID) = 1 THEN
            CASE
                WHEN OldRow.TagChoiceID != '_' THEN // NOT DEFAULT TagChoice so
ONLY People WITH related TagValue records need updating
                    UPDATE PersonKeyWords pkw SET TagRefreshRequired = 1 FROM
PersonKeyWords pkw JOIN TagValue tv ON tv.TagLocation = OldRow.TagLocation
AND tv.TagID = OldRow.TagID AND tv.TagChoiceID = OldRow.TagChoiceID AND
tv.ID = pkw.PersonID
                WHEN OldRow.TagLocation LIKE 'A%' THEN // ONLY need TO SET VALUES
FOR those registered WITH that department
                    UPDATE PersonKeyWords pkw SET TagRefreshRequired = 1 FROM
PersonKeyWords pkw JOIN SEARCH s ON s.PersonID = pkw.PersonID AND
s.DepartmentID = SUBSTRING(OldRow.TagLocation,2)
                WHEN NewRow.TagLocation = 'P' THEN // need TO SET VALUES FOR ALL
                    UPDATE PersonKeyWords pkw SET TagRefreshRequired = 1
            END CASE;

        WHEN (OldRow.TagLocation LIKE 'C%' ) AND (SELECT IncludeInKeywordSearch
FROM Tag WHERE Tag.TagLocation = OldRow.TagLocation AND Tag.TagID =
OldRow.TagID) = 1 THEN
            CASE
                WHEN OldRow.TagChoiceID != '_' THEN // NOT DEFAULT TagChoice so
ONLY People WITH related TagValue records need updating
```



```
        UPDATE CompanyKeyWords pkw SET TagRefreshRequired = 1 FROM
CompanyKeyWords pkw JOIN TagValue tv ON tv.TagLocation = OldRow.TagLocation
AND tv.TagID = OldRow.TagID AND tv.TagChoiceID = OldRow.TagChoiceID AND
tv.ID = pkw.CompanyID
        WHEN LENGTH(OldRow.TagLocation)>1 THEN // ONLY need TO SET VALUES
FOR those registered WITH that department
        UPDATE CompanyKeyWords pkw SET TagRefreshRequired = 1 FROM
CompanyKeyWords pkw JOIN Company c ON (c.CompanyID = pkw.CompanyID), Company
c KEY JOIN staff s WHERE s.Defaultdepartid = SUBSTRING(OldRow.TagLocation,2)
        WHEN NewRow.TagLocation = 'C' THEN // need TO SET VALUES FOR ALL
        UPDATE CompanyKeyWords pkw SET TagRefreshRequired = 1
    END CASE;

    WHEN (OldRow.TagLocation LIKE 'V%' ) AND (SELECT IncludeInKeywordSearch
FROM Tag WHERE Tag.TagLocation = OldRow.TagLocation AND Tag.TagID =
OldRow.TagID) = 1 THEN
        CASE
            WHEN OldRow.TagChoiceID != '_' THEN // NOT DEFAULT TagChoice so
ONLY People WITH related TagValue records need updating
            UPDATE VacancyKeyWords pkw SET TagRefreshRequired = 1 FROM
VacancyKeyWords pkw JOIN TagValue tv ON tv.TagLocation = OldRow.TagLocation
AND tv.TagID = OldRow.TagID AND tv.TagChoiceID = OldRow.TagChoiceID AND
tv.ID = pkw.VacancyID
            WHEN LENGTH(OldRow.TagLocation)>1 THEN // ONLY need TO SET VALUES
FOR those registered WITH that department
            UPDATE VacancyKeyWords pkw SET TagRefreshRequired = 1 FROM
VacancyKeyWords pkw JOIN Vacancy s ON s.VacancyID = pkw.VacancyID AND
s.DepartmentID = SUBSTRING(OldRow.TagLocation,2)
            WHEN NewRow.TagLocation = 'P' THEN // need TO SET VALUES FOR ALL
            UPDATE VacancyKeyWords pkw SET TagRefreshRequired = 1
        END CASE;
    // WHEN ..... - FOR other tag locations
    END CASE;

    WHEN INSERTING THEN
        CASE WHEN (NewRow.TagLocation = 'P' OR NewRow.TagLocation LIKE
'A%' ) AND (SELECT IncludeInKeywordSearch FROM Tag WHERE Tag.TagLocation =
NewRow.TagLocation AND Tag.TagID = NewRow.TagID) = 1 THEN
            IF NewRow.TagChoiceID = '_' // we are ONLY concerned WITH NEW
DEFAULT choices - anything ELSE will be dealt WITH BY the TagValue TRIGGER
            THEN
                IF NewRow.TagLocation LIKE 'A%' THEN // ONLY need TO SET
VALUES FOR those registered WITH 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)
                ELSE // everyone will need the TagKeyWords re-evaluated
                    UPDATE PersonKeyWords pkw SET TagRefreshRequired =
```



```
1
        END IF;
    END IF;
    WHEN ( NewRow.TagLocation LIKE 'C%' ) AND (SELECT IncludeInKeywordSearch
FROM Tag WHERE Tag.TagLocation = NewRow.TagLocation AND Tag.TagID =
NewRow.TagID) = 1 THEN
        IF NewRow.TagChoiceID = '_' // we are ONLY concerned WITH NEW
DEFAULT choices - anything ELSE will be dealt WITH BY the TagValue TRIGGER
        THEN
            IF LENGTH(NewRow.TagLocation)>1 THEN // ONLY need TO SET
VALUES FOR those registered WITH that department
                UPDATE CompanyKeyWords pkw SET TagRefreshRequired
= 1 FROM CompanyKeyWords pkw JOIN Company c ON (c.CompanyID =
pkw.CompanyID), company c KEY JOIN staff s WHERE s.Defaultdeptid =
SUBSTRING(NewRow.TagLocation,2)
            ELSE // everyone will need the TagKeyWords re-evaluated
                UPDATE CompanyKeyWords pkw SET TagRefreshRequired
= 1
            END IF;
        END IF;
    WHEN ( NewRow.TagLocation LIKE 'V%' ) AND (SELECT IncludeInKeywordSearch
FROM Tag WHERE Tag.TagLocation = NewRow.TagLocation AND Tag.TagID =
NewRow.TagID) = 1 THEN
        IF NewRow.TagChoiceID = '_' // we are ONLY concerned WITH NEW
DEFAULT choices - anything ELSE will be dealt WITH BY the TagValue TRIGGER
        THEN
            IF LENGTH(NewRow.TagLocation)>1 THEN // ONLY need TO SET
VALUES FOR those registered WITH 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)
            ELSE // everyone will need the TagKeyWords re-evaluated
                UPDATE VacancyKeyWords pkw SET TagRefreshRequired
= 1
            END IF;
        END IF;
    END CASE;
    WHEN UPDATING THEN
        IF NewRow.Description != OldRow.Description AND // we are ONLY
concerned IF the Description has changed
        (NewRow.TagLocation = 'P' OR NewRow.TagLocation LIKE 'A%' ) AND
        (SELECT IncludeInKeywordSearch FROM Tag WHERE Tag.TagLocation
= NewRow.TagLocation AND Tag.TagID = NewRow.TagID) = 1 THEN
            CASE
                WHEN OldRow.TagChoiceID != '_' THEN // NOT DEFAULT
TagChoice so ONLY People WITH related TagValue records need updating
                    UPDATE PersonKeyWords pkw SET TagRefreshRequired = 1
FROM PersonKeyWords pkw JOIN TagValue tv ON tv.TagLocation =
```



```
OldRow.TagLocation AND tv.TagID = OldRow.TagID AND tv.TagChoiceID =
OldRow.TagChoiceID AND tv.ID = pkw.PersonID
    WHEN NewRow.TagLocation LIKE 'A%' THEN // we ONLY need TO
deal WITH the 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;
    ELSE IF NewRow.Description != OldRow.Description AND // we are
ONLY concerned IF the Description has changed
        (NewRow.TagLocation LIKE 'C%' ) AND
        (SELECT IncludeInKeywordSearch FROM Tag WHERE Tag.TagLocation
= NewRow.TagLocation AND Tag.TagID = NewRow.TagID) = 1 THEN
        CASE
            WHEN OldRow.TagChoiceID != '_' THEN // NOT DEFAULT
TagChoice so ONLY People WITH related TagValue records need updating
                UPDATE CompanyKeyWords pkw SET TagRefreshRequired = 1
FROM CompanyKeyWords pkw JOIN TagValue tv ON tv.TagLocation =
OldRow.TagLocation AND tv.TagID = OldRow.TagID AND tv.TagChoiceID =
OldRow.TagChoiceID AND tv.ID = pkw.CompanyID
            WHEN NewRow.TagLocation LIKE 'A%' THEN // we ONLY need TO
deal WITH the department
                UPDATE CompanyKeyWords pkw SET TagRefreshRequired = 1
FROM CompanyKeyWords pkw JOIN Company s ON (s.CompanyID = pkw.CompanyID),
Company s KEY JOIN Staff s WHERE s.Defaultdepartid =
SUBSTRING(NewRow.TagLocation,2)
            WHEN NewRow.TagLocation = 'C' THEN // need TO SET VALUES
FOR ALL
                UPDATE CompanyKeyWords pkw SET TagRefreshRequired = 1
            END CASE;
            ELSE IF NewRow.Description != OldRow.Description AND // we are
ONLY concerned IF the Description has changed
                ( NewRow.TagLocation LIKE 'V%' ) AND
                (SELECT IncludeInKeywordSearch FROM Tag WHERE Tag.TagLocation
= NewRow.TagLocation AND Tag.TagID = NewRow.TagID) = 1 THEN
                CASE
                    WHEN OldRow.TagChoiceID != '_' THEN // NOT DEFAULT
TagChoice so ONLY People WITH related TagValue records need updating
                        UPDATE VacancyKeyWords pkw SET TagRefreshRequired = 1
FROM VacancyKeyWords pkw JOIN TagValue tv ON tv.TagLocation =
OldRow.TagLocation AND tv.TagID = OldRow.TagID AND tv.TagChoiceID =
OldRow.TagChoiceID AND tv.ID = pkw.Vacancyid
                    WHEN LENGTH(NewRow.TagLocation)>1 THEN // we ONLY need TO
deal WITH the 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)
      WHEN NewRow.TagLocation = 'V' THEN // need TO SET VALUES
FOR ALL
      UPDATE VacancyKeyWords pkw SET TagRefreshRequired = 1
      END CASE;
      END IF END IF END IF ;
END CASE
END
}
GO
```

From:

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

Permanent link:

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

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

