



pears.diary



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

Description

Appointment diary records.

Columns

Column	Type	Null	Default	Comment
diaryid	char(20)	NOT NULL		
staffid	char(20)	NOT NULL		
personid	char(20)	NULL		
vacancyid	char(20)	NULL		
employmentid	char(20)	NULL		
progressid	char(20)	NULL		
placementid	char(20)	NULL		
diaryfrom	timestamp	NOT NULL		
diaryto	timestamp	NULL		
description	char(100)	NULL		
notes	long varchar	NULL		
DurationType	char(1)	NULL		R=Reminder, M=Minutes, H=Hours, D=Days, W=Weeks
Duration	double	NULL		
PopupMinutes	smallint	NULL		
PopupTime	timestamp	NULL		
WhenEntered	timestamp	NULL	current timestamp	
WhoEntered	char(20)	NULL		Auto-entered staffid, but no ref integ because it would break numerous key joins by making them ambiguous
OriginalDiaryID	char(20)	NULL		Used to prevent circular escalations
groupid	char(20)	NULL		link multiple staff to diary
RecurDays	integer	NULL	0	
priority	smallint	NULL	5	1 - high, 5 - low
CanEscalate	integer	NULL		



Column	Type	Null	Default	Comment
ICSFileUID	char(20)	NULL		
collectionid	char(20)	NULL		
ContactEventID	char(20)	NULL		

Primary Key

- diaryid

Foreign Keys

Constraint	Columns	References	Delete/update action
staff	staffid	pears.staff (staffid)	NOT NULL;
person	personid	pears.Person (personid)	ON DELETE SET NULL
vacancy	vacancyid	pears.vacancy (vacancyid)	ON DELETE SET NULL
employment	employmentid	pears.employment (employmentid)	ON DELETE SET NULL
progress	progressid	pears.progress (progressid)	ON DELETE SET NULL
placement	placementid	pears.Placement (placementid)	ON DELETE SET NULL
collection	collectionid	pears.Collection (CollectionID)	ON DELETE SET NULL
contactevent	ContactEventID	pears.contactevent (contacteventid)	ON DELETE CASCADE

Referenced By

Table	Constraint	Columns	Referenced columns
pears.IQXNetNotificationCascade	Diary	DiaryID	diaryid

Indexes

Name	Type	Columns	Detail
diary_from	Index	diaryfrom	
diary_stafffrom	Index	staffid, diaryfrom	
diary_to	Index	diaryto	
diary_staffto	Index	staffid, diaryto	
diary_popup	Index	PopupTime	
diary_whenentered	Index	WhenEntered	

Triggers

Name	Timing	Event
Diary_Insert	before	insert order 5
Diary_Insert_2	before	insert order 6
Diary_Update	before	update order 1



Name	Timing	Event
Diary_Update_After_TimeSynch	after	update of "diaryfrom", "diaryto" order 3

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."diary"
-- Table comment: Appointment diary records.
-- Statement count: 29

CREATE TABLE "pears"."diary" (
    "diaryid" CHAR(20) NOT NULL
    , "staffid" CHAR(20) NOT NULL
    , "personid" CHAR(20) NULL
    , "vacancyid" CHAR(20) NULL
    , "employmentid" CHAR(20) NULL
    , "progressid" CHAR(20) NULL
    , "placementid" CHAR(20) NULL
    , "diaryfrom" TIMESTAMP NOT NULL
    , "diaryto" TIMESTAMP NULL
    , "description" CHAR(100) NULL
    , "notes" long VARCHAR NULL
    , "DurationType" CHAR(1) NULL
    , "Duration" DOUBLE NULL
    , "PopupMinutes" SMALLINT NULL
    , "PopupTime" TIMESTAMP NULL COMPUTE
    ("dateadd"("minute", 0 - "PopupMinutes", "DiaryFrom"))
    , "WhenEntered" TIMESTAMP NULL DEFAULT CURRENT
TIMESTAMP
    , "WhoEntered" CHAR(20) NULL
    , "OriginalDiaryID" CHAR(20) NULL
    , "groupid" CHAR(20) NULL
    , "RecurDays" INTEGER NULL DEFAULT 0
    , "priority" SMALLINT NULL DEFAULT 5
    , "CanEscalate" INTEGER NULL
    , "ICSFileUID" CHAR(20) NULL
    , "collectionid" CHAR(20) NULL
    , "ContactEventID" CHAR(20) NULL
    , PRIMARY KEY ("diaryid" ASC)
)
GO

COMMENT ON COLUMN "pears"."diary"."DurationType" IS
    'R=Reminder, M=Minutes, H=Hours, D=Days, W=Weeks'
```



GO

```
COMMENT ON COLUMN "pears"."diary"."WhoEntered" IS
    'Auto-entered staffid, but no ref integ because it would break numerous
    key joins by making them ambiguous'
```

GO

```
COMMENT ON COLUMN "pears"."diary"."OriginalDiaryID" IS
    'Used to prevent circular escalations'
```

GO

```
COMMENT ON COLUMN "pears"."diary"."groupid" IS
    'link multiple staff to diary'
```

GO

```
COMMENT ON COLUMN "pears"."diary"."priority" IS
    '1 - high, 5 - low'
```

GO

```
COMMENT ON TABLE "pears"."diary" IS
    'Appointment diary records.'
```

GO

```
ALTER TABLE "pears"."diary"
    ADD NOT NULL FOREIGN KEY "staff" ("staffid" ASC)
    REFERENCES "pears"."staff" ("staffid")
```

GO

```
ALTER TABLE "pears"."diary"
    ADD FOREIGN KEY "person" ("personid" ASC)
    REFERENCES "pears"."Person" ("personid")
    ON DELETE SET NULL
```

GO

```
ALTER TABLE "pears"."diary"
    ADD FOREIGN KEY "vacancy" ("vacancyid" ASC)
    REFERENCES "pears"."vacancy" ("vacancyid")
    ON DELETE SET NULL
```

GO



```
ALTER TABLE "pears"."diary"  
  ADD FOREIGN KEY "employment" ("employmentid" ASC)  
  REFERENCES "pears"."employment" ("employmentid")  
  ON DELETE SET NULL
```

```
GO
```

```
ALTER TABLE "pears"."diary"  
  ADD FOREIGN KEY "progress" ("progressid" ASC)  
  REFERENCES "pears"."progress" ("progressid")  
  ON DELETE SET NULL
```

```
GO
```

```
ALTER TABLE "pears"."diary"  
  ADD FOREIGN KEY "placement" ("placementid" ASC)  
  REFERENCES "pears"."Placement" ("placementid")  
  ON DELETE SET NULL
```

```
GO
```

```
ALTER TABLE "pears"."diary"  
  ADD FOREIGN KEY "collection" ("collectionid" ASC)  
  REFERENCES "pears"."Collection" ("CollectionID")  
  ON DELETE SET NULL
```

```
GO
```

```
ALTER TABLE "pears"."diary"  
  ADD FOREIGN KEY "contactevent" ("ContactEventID" ASC)  
  REFERENCES "pears"."contactevent" ("contacteventid")  
  ON DELETE CASCADE
```

```
GO
```

```
CREATE INDEX "diary_from" ON "pears"."diary"  
  ( "diaryfrom" )
```

```
GO
```

```
CREATE INDEX "diary_stafffrom" ON "pears"."diary"  
  ( "staffid","diaryfrom" )
```

```
GO
```

```
CREATE INDEX "diary_to" ON "pears"."diary"  
  ( "diaryto" )
```



GO

```
CREATE INDEX "diary_staffto" ON "pears"."diary"  
  ( "staffid","diaryto" )
```

GO

```
CREATE INDEX "diary_popup" ON "pears"."diary"  
  ( "PopupTime" )
```

GO

```
CREATE INDEX "diary_whenentered" ON "pears"."diary"  
  ( "WhenEntered" )
```

GO

```
CREATE TRIGGER "Diary_Insert" BEFORE INSERT ORDER 5 ON  
"pears"."diary"  
REFERENCING NEW AS "new_diary"  
FOR each ROW  
WHEN("new_diary"."whoentered" IS NULL)  
BEGIN  
  SET "new_diary"."whoentered" = "userstaffid"  
exception  
  WHEN others THEN  
    SET "new_diary"."whoentered" = NULL  
END  
GO
```

```
COMMENT TO PRESERVE FORMAT ON TRIGGER "pears"."diary"."Diary_Insert" IS  
{CREATE TRIGGER Diary_Insert  
  BEFORE INSERT ORDER 5 ON  
pears.diary  
REFERENCING NEW AS new_diary  
FOR each ROW  
WHEN(new_diary.whoentered IS NULL)  
BEGIN  
  SET new_diary.whoentered=userstaffid  
exception  
  WHEN others THEN  
    SET new_diary.whoentered=NULL  
END  
}  
GO
```



```
CREATE TRIGGER "Diary_Insert_2" BEFORE INSERT ORDER 6 ON
"pears"."diary"
REFERENCING NEW AS "new_diary"
FOR each ROW
BEGIN
  IF "new_diary"."durationtype" IN( 'M','H','D','W','R' ) THEN
    IF "isnull"("new_diary"."duration",0) = 0 THEN
      SET "new_diary"."durationtype" = 'R';
      SET "new_diary"."duration" = 0.0
    END IF;
    CASE "new_diary"."durationtype" WHEN 'M' THEN
      SET "new_diary"."diaryto" =
"dateadd"("minute","new_diary"."duration","new_diary"."diaryfrom") WHEN 'H'
THEN
      SET "new_diary"."diaryto" =
"dateadd"("minute",("new_diary"."duration")*60,"new_diary"."diaryfrom") WHEN
'D' THEN
      SET "new_diary"."diaryto" =
"dateadd"("hour",("new_diary"."duration")*24,"new_diary"."diaryfrom") WHEN
'W' THEN
      SET "new_diary"."diaryto" =
"dateadd"("day",("new_diary"."duration")*7,"new_diary"."diaryfrom") ELSE
      SET "new_diary"."diaryto" = "new_diary"."diaryfrom"
    END CASE
  ELSE
    IF "new_diary"."diaryto" IS NULL THEN
      SET "new_diary"."diaryto" = "new_diary"."diaryfrom"
    END IF;
    IF "datediff"("minute","new_diary"."diaryfrom","new_diary"."diaryto") <
1 THEN
      SET "new_diary"."durationtype" = 'R';
      SET "new_diary"."duration" = 0
    ELSEIF "datediff"("hour","new_diary"."diaryfrom","new_diary"."diaryto")
< 1 THEN
      SET "new_diary"."durationtype" = 'M';
      SET "new_diary"."duration" =
"datediff"("minute","new_diary"."diaryfrom","new_diary"."diaryto")
    ELSEIF "datediff"("day","new_diary"."diaryfrom","new_diary"."diaryto") <
1 THEN
      SET "new_diary"."durationtype" = 'H';
      SET "new_diary"."duration" =
"round"("datediff"("minute","new_diary"."diaryfrom","new_diary"."diaryto")/6
0.0,2)
    ELSE SET "new_diary"."durationtype" = 'D';
      SET "new_diary"."duration" =
"round"("datediff"("hour","new_diary"."diaryfrom","new_diary"."diaryto")/24.
0,2)
```



```
END IF
END IF
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER "pears"."diary"."Diary_Insert_2" IS
{CREATE TRIGGER Diary_Insert_2
  BEFORE INSERT ORDER 6 ON
pears.diary
REFERENCING NEW AS new_diary
FOR each ROW
BEGIN
  IF new_diary.durationtype IN( 'M','H','D','W','R') THEN
    IF isnull(new_diary.duration,0) = 0 THEN
      SET new_diary.durationtype='R';
      SET new_diary.duration=0.0
    END IF;
    CASE new_diary.durationtype WHEN 'M' THEN
      SET
new_diary.diaryto=dateadd(MINUTE,new_diary.duration,new_diary.diaryfrom)
    WHEN 'H' THEN
      SET
new_diary.diaryto=dateadd(MINUTE,(new_diary.duration)*60,new_diary.diaryfrom
) WHEN 'D' THEN
      SET
new_diary.diaryto=dateadd(HOUR,(new_diary.duration)*24,new_diary.diaryfrom)
    WHEN 'W' THEN
      SET
new_diary.diaryto=dateadd(DAY,(new_diary.duration)*7,new_diary.diaryfrom)
    ELSE
      SET new_diary.diaryto=new_diary.diaryfrom
    END CASE
  ELSE
    IF new_diary.diaryto IS NULL THEN
      SET new_diary.diaryto=new_diary.diaryfrom
    END IF;
    IF datediff(MINUTE,new_diary.diaryfrom,new_diary.diaryto) < 1 THEN
      SET new_diary.durationtype='R';
      SET new_diary.duration=0
    ELSEIF datediff(HOUR,new_diary.diaryfrom,new_diary.diaryto) < 1 THEN
      SET new_diary.durationtype='M';
      SET
new_diary.duration=datediff(MINUTE,new_diary.diaryfrom,new_diary.diaryto)
    ELSEIF datediff(DAY,new_diary.diaryfrom,new_diary.diaryto) < 1 THEN
      SET new_diary.durationtype='H';
      SET
new_diary.duration=round(datediff(MINUTE,new_diary.diaryfrom,new_diary.diary
```



```
to)/60.0,2)
    ELSE SET new_diary.durationtype='D';
    SET
new_diary.duration=round(datediff(HOUR,new_diary.diaryfrom,new_diary.diaryto
)/24.0,2)
    END IF
END IF
END
}
GO

CREATE TRIGGER "Diary_Update" BEFORE UPDATE ORDER 1 ON
"pears"."diary"
REFERENCING OLD AS "old_diary" NEW AS "new_diary"
FOR each ROW
BEGIN
    IF "new_diary"."durationtype" IN( 'M','H','D','W','R' ) THEN
        IF "isnull"("new_diary"."duration",0) = 0 THEN
            SET "new_diary"."durationtype" = 'R';
            SET "new_diary"."duration" = 0.0
        END IF;
        CASE "new_diary"."durationtype" WHEN 'M' THEN
            SET "new_diary"."diaryto" =
            "dateadd"("minute","new_diary"."duration","new_diary"."diaryfrom") WHEN 'H'
THEN
            SET "new_diary"."diaryto" =
            "dateadd"("minute",("new_diary"."duration")*60,"new_diary"."diaryfrom") WHEN
'D' THEN
            SET "new_diary"."diaryto" =
            "dateadd"("hour",("new_diary"."duration")*24,"new_diary"."diaryfrom") WHEN
'W' THEN
            SET "new_diary"."diaryto" =
            "dateadd"("day",("new_diary"."duration")*7,"new_diary"."diaryfrom") ELSE
            SET "new_diary"."diaryto" = "new_diary"."diaryfrom"
        END CASE
    ELSE
        IF "new_diary"."diaryto" IS NULL THEN
            SET "new_diary"."diaryto" = "new_diary"."diaryfrom"
        END IF;
        IF "datediff"("minute","new_diary"."diaryfrom","new_diary"."diaryto") <
1 THEN
            SET "new_diary"."durationtype" = 'R';
            SET "new_diary"."duration" = 0
        ELSEIF "datediff"("hour","new_diary"."diaryfrom","new_diary"."diaryto")
< 1 THEN
            SET "new_diary"."durationtype" = 'M';
            SET "new_diary"."duration" =
```



```
"datediff"("minute","new_diary"."diaryfrom","new_diary"."diaryto")
ELSEIF "datediff"("day","new_diary"."diaryfrom","new_diary"."diaryto") <
1 THEN
    SET "new_diary"."durationtype" = 'H';
    SET "new_diary"."duration" =
"round"("datediff"("minute","new_diary"."diaryfrom","new_diary"."diaryto")/6
0.0,2)
    ELSE SET "new_diary"."durationtype" = 'D';
    SET "new_diary"."duration" =
"round"("datediff"("hour","new_diary"."diaryfrom","new_diary"."diaryto")/24.
0,2)
    END IF END IF
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER "pears"."diary"."Diary_Update" IS
{CREATE TRIGGER Diary_Update
BEFORE UPDATE ORDER 1 ON
pears.diary
REFERENCING OLD AS old_diary NEW AS new_diary
FOR each ROW
BEGIN
    IF new_diary.durationtype IN( 'M','H','D','W','R') THEN
        IF isnull(new_diary.duration,0) = 0 THEN
            SET new_diary.durationtype='R';
            SET new_diary.duration=0.0
        END IF;
        CASE new_diary.durationtype WHEN 'M' THEN
            SET
new_diary.diaryto=dateadd(MINUTE,new_diary.duration,new_diary.diaryfrom)
        WHEN 'H' THEN
            SET
new_diary.diaryto=dateadd(MINUTE,(new_diary.duration)*60,new_diary.diaryfrom
) WHEN 'D' THEN
            SET
new_diary.diaryto=dateadd(HOUR,(new_diary.duration)*24,new_diary.diaryfrom)
        WHEN 'W' THEN
            SET
new_diary.diaryto=dateadd(DAY,(new_diary.duration)*7,new_diary.diaryfrom)
        ELSE
            SET new_diary.diaryto=new_diary.diaryfrom
        END CASE
    ELSE
        IF new_diary.diaryto IS NULL THEN
            SET new_diary.diaryto=new_diary.diaryfrom
        END IF;
        IF datediff(MINUTE,new_diary.diaryfrom,new_diary.diaryto) < 1 THEN
```



```
    SET new_diary.durationtype='R';
    SET new_diary.duration=0
ELSEIF datediff(HOUR,new_diary.diaryfrom,new_diary.diaryto) < 1 THEN
    SET new_diary.durationtype='M';
    SET
new_diary.duration=datediff(MINUTE,new_diary.diaryfrom,new_diary.diaryto)
    ELSEIF datediff(DAY,new_diary.diaryfrom,new_diary.diaryto) < 1 THEN
    SET new_diary.durationtype='H';
    SET
new_diary.duration=round(datediff(MINUTE,new_diary.diaryfrom,new_diary.diaryto)/60.0,2)
    ELSE SET new_diary.durationtype='D';
    SET
new_diary.duration=round(datediff(HOUR,new_diary.diaryfrom,new_diary.diaryto)/24.0,2)
    END IF
END IF;
END
}
GO
```

```
CREATE TRIGGER "Diary_Update_After_TimeSynch" after UPDATE OF "diaryfrom",
"diaryto" ORDER 3 ON
"pears"."diary"
REFERENCING OLD AS "old_diary" NEW AS "new_diary"
FOR each ROW
BEGIN
    IF "isnull"("new_diary"."groupid",'') <> '' THEN
        UPDATE "diary" SET "diaryfrom" = "new_diary"."diaryfrom","diaryto" =
"new_diary"."diaryto" WHERE "groupid"
            = "new_diary"."groupid" AND "diaryid" <> "new_diary"."diaryid"
    END IF
END
GO
```

```
COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."diary"."Diary_Update_After_TimeSynch" IS
{CREATE TRIGGER Diary_Update_After_TimeSynch
after UPDATE OF diaryfrom, diaryto ORDER 3 ON
pears.diary
REFERENCING OLD AS old_diary NEW AS new_diary
FOR each ROW
BEGIN
    IF isnull(new_diary.groupid,'') <> '' THEN
UPDATE diary SET diaryfrom = new_diary.diaryfrom, diaryto =
new_diary.diaryto WHERE groupid
```



```
= new_diary.groupid AND diaryid <> new_diary.diaryid  
  END IF  
END  
}  
GO
```

From:

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

Permanent link:

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

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

