



pears.progress



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

Description

Progress state of candidates under consideration / placed / rejected in vacancies. Links candidates to vacancies.

Columns

Column	Type	Null	Default	Comment
progressid	char(20)	NOT NULL		
vacancyid	char(20)	NOT NULL		
status	char(4)	NOT NULL		
note	long varchar	NULL		
actiondate	date	NULL		
rejectionreason	char(100)	NULL		
staffid	char(20)	NOT NULL		
personid	char(20)	NOT NULL		
placementid	char(20)	NULL		
InterviewDate	date	NULL		
InterviewTime	time	NULL		
actiontime	time	NULL		
priority	smallint	NULL	5	1 - high, 5 - low
noreemploy	smallint	NULL		
score	smallint	NULL		
defunct	tinyint	NULL		

Primary Key

- progressid

Foreign Keys



Constraint	Columns	References	Delete/update action
staff	staffid	pears.staff (staffid)	NOT NULL;
person	personid	pears.Person (personid)	NOT NULL; ON DELETE CASCADE
placement	placementid	pears.Placement (placementid)	ON DELETE SET NULL
vacancy	vacancyid	pears.vacancy (vacancyid)	NOT NULL;

Referenced By

Table	Constraint	Columns	Referenced columns
pears.contactevent	progress	progressid	progressid
pears.diary	progress	progressid	progressid
pears.ProgressHistory	Progress	ProgressID	progressid
pears.SentCVs	progress	progressid	progressid

Indexes

Name	Type	Columns	Detail
progress_actiondate	Index	actiondate	
progressstatus	Index	status	
progress_notestext	Text index	note	CONFIGURATION "SYS"."default_char" IMMEDIATE REFRESH

Triggers

Name	Timing	Event
ProgressInsert	after	insert order 1
ProgressUpdate	after	update of "status", "InterviewDate", "InterviewTime", "Note" order 1

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."progress"
-- Table comment: Progress state of candidates under consideration / placed
-- / rejected in vacancies. Links candidates to vacancies.
-- Statement count: 14

CREATE TABLE "pears"."progress" (
    "progressid"          CHAR(20) NOT NULL
    ,"vacancyid"          CHAR(20) NOT NULL
    ,"status"             CHAR(4) NOT NULL
    ,"note"               long VARCHAR NULL
    ,"actiondate"         DATE NULL
    ,"rejectionreason"    CHAR(100) NULL
)
```



```
, "staffid"          CHAR(20) NOT NULL
, "personid"         CHAR(20) NOT NULL
, "placementid"      CHAR(20) NULL
, "InterviewDate"    DATE NULL
, "InterviewTime"    TIME NULL
, "actiontime"       TIME NULL
, "priority"         SMALLINT NULL DEFAULT 5
, "noreemploy"       SMALLINT NULL
, "score"            SMALLINT NULL
, "defunct"          tinyint NULL
, PRIMARY KEY ("progressid" ASC)
)
GO

COMMENT ON COLUMN "pears"."progress"."priority" IS
    '1 - high, 5 - low'
GO

COMMENT ON TABLE "pears"."progress" IS
    'Progress state of candidates under consideration / placed / rejected in
    vacancies. Links candidates to vacancies.'
GO

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

ALTER TABLE "pears"."progress"
    ADD NOT NULL FOREIGN KEY "person" ("personid" ASC)
    REFERENCES "pears"."Person" ("personid")
    ON DELETE CASCADE
GO

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

ALTER TABLE "pears"."progress"
    ADD NOT NULL FOREIGN KEY "vacancy" ("vacancyid" ASC)
```



```
REFERENCES "pears"."vacancy" ("vacancyid")
GO

CREATE INDEX "progress_actiondate" ON "pears"."progress"
( "actiondate" DESC )
GO

CREATE INDEX "progressstatus" ON "pears"."progress"
( "status" )
GO

CREATE TEXT INDEX "progress_notestext" ON "pears"."progress"
( "note" ) CONFIGURATION "SYS"."default_char" IMMEDIATE REFRESH
GO

CREATE TRIGGER "ProgressInsert" after INSERT ORDER 1 ON
"pears"."progress"
REFERENCING NEW AS "new_prog"
FOR each ROW
BEGIN
    DECLARE @ContactEventClass CHAR(2);
    DECLARE @ProgressName CHAR(40);
    INSERT INTO "progresshistory"(
        "progresshistoryid","progressid","status","historydate","historytime","staff
        id","InterviewDate","InterviewTime","Note" ) VALUES
        (
            "uniquekey"("new_prog"."progressid"),"new_prog"."progressid","new_prog"."sta
            tus",CURRENT DATE,CURRENT
            TIME,"new_prog"."staffid","new_prog"."InterviewDate","new_prog"."InterviewTi
            me","new_prog"."Note" ) ;
    SELECT "ContactEventClass","Name" INTO @ContactEventClass,@ProgressName
    FROM "Status" WHERE "Status"."Type" = 'R' AND "new_prog"."Status" =
    "Status"."Status";
    IF @ContactEventClass IS NOT NULL THEN
        INSERT INTO "ContactEvent"(
            "ContactEventID","ClassCode","Personid","staffid","vacancyid","employmentid"
            ,"progressid","contactdate","contacttime","description","who","notes","Divis
            ionID" )
            SELECT "uniquekey"('X'),
                @ContactEventClass,
                "new_prog"."PersonID",
                "new_prog"."StaffID",
                "new_prog"."VacancyID",
                (SELECT "EmploymentID" FROM "Vacancy" WHERE "Vacancy"."VacancyID" =
```



```
"new_prog"."VacancyID"),
    "new_prog"."ProgressID",
    CURRENT DATE,
    CURRENT TIME,
    @ProgressName, 'A', 'Automatically generated from Progress record',
    (SELECT "DivisionID" FROM "Company" KEY JOIN "Employment" KEY JOIN
"Vacancy" WHERE "Vacancy"."VacancyID" = "new_prog"."VacancyID" AND
"Company"."CompanyID" = "Employment"."CompanyID")
    END IF
END
GO

COMMENT TO PRESERVE FORMAT ON TRIGGER "pears"."progress"."ProgressInsert" IS
{CREATE TRIGGER ProgressInsert
  after INSERT ORDER 1 ON
pears.progress
REFERENCING NEW AS new_prog
FOR each ROW
BEGIN
  DECLARE @ContactEventClass CHAR(2);
  DECLARE @ProgressName CHAR(40);
  INSERT INTO progresshistory(
progresshistoryid, progressid, STATUS, historydate, historytime, staffid, InterviewDate, InterviewTime, Note) VALUES(
uniquekey(new_prog.progressid), new_prog.progressid, new_prog.status, CURRENT
DATE, CURRENT
TIME, new_prog.staffid, new_prog.InterviewDate, new_prog.InterviewTime, new_prog
.Note) ;
  SELECT ContactEventClass, Name INTO @ContactEventClass, @ProgressName FROM
STATUS WHERE STATUS.Type = 'R' AND new_prog.Status = STATUS.Status;
  IF @ContactEventClass IS NOT NULL THEN
    INSERT INTO ContactEvent(
ContactEventID, ClassCode, Personid, staffid, vacancyid, employmentid, progressid,
contactdate, contacttime, description, who, notes, DivisionID)
    SELECT uniquekey('X'),
    @ContactEventClass,
    new_prog.PersonID,
    new_prog.StaffID,
    new_prog.VacancyID,
    (SELECT EmploymentID FROM Vacancy WHERE Vacancy.VacancyID =
new_prog.VacancyID),
    new_prog.ProgressID,
    CURRENT DATE,
    CURRENT TIME,
    @ProgressName, 'A', 'Automatically generated from Progress record',
    (SELECT DivisionID FROM Company KEY JOIN Employment KEY JOIN Vacancy
WHERE Vacancy.VacancyID = new_prog.VacancyID AND Company.CompanyID =
```



```
Employment.CompanyID)
    END IF
END
}
GO

CREATE TRIGGER "ProgressUpdate" after UPDATE OF "status",
"InterviewDate", "InterviewTime",
"Note" ORDER 1 ON "pears"."progress"
REFERENCING NEW AS "new_prog"
FOR each ROW
BEGIN
    DECLARE @ContactEventClass CHAR(2);
    DECLARE @ProgressName CHAR(40);
    INSERT INTO "progresshistory"(
"progresshistoryid", "progressid", "status", "historydate", "historytime", "staff
id", "InterviewDate", "InterviewTime", "Note" ) VALUES
(
"uniquekey"("new_prog"."progressid"), "new_prog"."progressid", "new_prog"."sta
tus", CURRENT DATE, CURRENT
TIME, "new_prog"."staffid", "new_prog"."InterviewDate", "new_prog"."InterviewTi
me", "new_prog"."Note" ) ;
    SELECT "ContactEventClass", "Name" INTO @ContactEventClass, @ProgressName
FROM "Status" WHERE "Status"."Type" = 'R' AND "new_prog"."Status" =
"Status"."Status";
    IF @ContactEventClass IS NOT NULL THEN
        INSERT INTO "ContactEvent"(
"ContactEventID", "ClassCode", "Personid", "staffid", "vacancyid", "employmentid"
, "progressid", "contactdate", "contacttime", "description", "who", "notes", "Divis
ionID" )
        SELECT "uniquekey"('X'),
        @ContactEventClass,
        "new_prog"."PersonID",
        "new_prog"."StaffID",
        "new_prog"."VacancyID",
        (SELECT "EmploymentID" FROM "Vacancy" WHERE "Vacancy"."VacancyID" =
"new_prog"."VacancyID"),
        "new_prog"."ProgressID",
        CURRENT DATE,
        CURRENT TIME,
        @ProgressName, 'A', 'Automatically generated from Progress record',
        (SELECT "DivisionID" FROM "Company" KEY JOIN "Employment" KEY JOIN
"Vacancy" WHERE "Vacancy"."VacancyID" = "new_prog"."VacancyID" AND
"Company"."CompanyID" = "Employment"."CompanyID")
    END IF
END
GO
```



```
COMMENT TO PRESERVE FORMAT ON TRIGGER "pears"."progress"."ProgressUpdate" IS
{CREATE TRIGGER ProgressUpdate
  after UPDATE OF STATUS,
  InterviewDate, InterviewTime,
  Note ORDER 1 ON pears.progress
  REFERENCING NEW AS new_prog
  FOR each ROW
  BEGIN
    DECLARE @ContactEventClass CHAR(2);
    DECLARE @ProgressName CHAR(40);
    INSERT INTO progresshistory(
progresshistoryid, progressid, STATUS, historydate, historytime, staffid, InterviewDate, InterviewTime, Note) VALUES(
uniquekey(new_prog.progressid), new_prog.progressid, new_prog.status, CURRENT
DATE, CURRENT
TIME, new_prog.staffid, new_prog.InterviewDate, new_prog.InterviewTime, new_prog
.Note) ;
    SELECT ContactEventClass, Name INTO @ContactEventClass, @ProgressName FROM
STATUS WHERE STATUS.Type = 'R' AND new_prog.Status = STATUS.Status;
    IF @ContactEventClass IS NOT NULL THEN
      INSERT INTO ContactEvent(
ContactEventID, ClassCode, Personid, staffid, vacancyid, employmentid, progressid,
contactdate, contacttime, description, who, notes, DivisionID)
      SELECT uniquekey('X'),
        @ContactEventClass,
        new_prog.PersonID,
        new_prog.StaffID,
        new_prog.VacancyID,
        (SELECT EmploymentID FROM Vacancy WHERE Vacancy.VacancyID =
new_prog.VacancyID),
        new_prog.ProgressID,
        CURRENT DATE,
        CURRENT TIME,
        @ProgressName, 'A', 'Automatically generated from Progress record',
        (SELECT DivisionID FROM Company KEY JOIN Employment KEY JOIN Vacancy
WHERE Vacancy.VacancyID = new_prog.VacancyID AND Company.CompanyID =
Employment.CompanyID)
      END IF
    END
  }
GO
```



From:

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

Permanent link:

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

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

