pears.PlacementAnalysis

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

Description

Analysis of placement by nominal code

Columns

Column	Type	Null	Default	Comment
PlacementAnalysisID	char(20)	NOT NULL		
PlacementID	char(20)	NOT NULL		
NominalCode	char(12)	NOT NULL		
Percentage	double	NULL		
FeeAmount	double	NULL		
CommissionPayable	double	NULL		
PeriodAuthorised	integer	NULL		NULL if not yet authorised
CurrentlyAuthorised	smallint	NULL	0	Non-zero if this is a currently authorised record
WhenAuthorised	timestamp	NULL	current timestamp	
StaffID	char(20)	NULL		Auto-entered staffid
PartCredit	smallint	NULL	0	Non-zero if this represents a part credit

Primary Key

- PlacementAnalysisID

Foreign Keys

Constraint	Columns	References	Delete/update action
Placement	PlacementID	pears.Placement (placementid)	NOT NULL; ON DELETE CASCADE

Referenced By

- No incoming foreign keys found.

Indexes

Name	Type	Columns	Detail
PlacementAnalysis_Period	Index	PeriodAuthorised	

Triggers

Name	Timing	Event
PlacementAnalysisInsert	before	insert order 1
PlacementAnalysisAuthorised	before	update of “CurrentlyAuthorised” order 1
placementanalysisauditupdate	before	update of “NominalCode”, “Percentage” order 2
placementanalysisauditinsdel	before	insert,delete order 1

Original SQL

```
-- IQX database structure split by table
-- Source: IQXDatabaseStructure - with comments.sql
-- Table: "pears"."PlacementAnalysis"
-- Table comment: Analysis of placement by nominal code
-- Statement count: 16

CREATE TABLE "pears"."PlacementAnalysis" (
    "PlacementAnalysisID"            CHAR(20) NOT NULL
   ,"PlacementID"                    CHAR(20) NOT NULL
   ,"NominalCode"                    CHAR(12) NOT NULL
   ,"Percentage"                     DOUBLE NULL
   ,"FeeAmount"                      DOUBLE NULL
   ,"CommissionPayable"              DOUBLE NULL
   ,"PeriodAuthorised"               INTEGER NULL
   ,"CurrentlyAuthorised"            SMALLINT NULL DEFAULT 0
   ,"WhenAuthorised"                 TIMESTAMP NULL DEFAULT CURRENT
TIMESTAMP
   ,"StaffID"                        CHAR(20) NULL
   ,"PartCredit"                     SMALLINT NULL DEFAULT 0
   ,PRIMARY KEY ("PlacementAnalysisID" ASC)
)
GO


COMMENT ON COLUMN "pears"."PlacementAnalysis"."PeriodAuthorised" IS
```

```
    'NULL if not yet authorised'
GO


COMMENT ON COLUMN "pears"."PlacementAnalysis"."CurrentlyAuthorised" IS
    'Non-zero if this is a currently authorised record'
GO


COMMENT ON COLUMN "pears"."PlacementAnalysis"."StaffID" IS
    'Auto-entered staffid'
GO


COMMENT ON COLUMN "pears"."PlacementAnalysis"."PartCredit" IS
    'Non-zero if this represents a part credit'
GO


COMMENT ON TABLE "pears"."PlacementAnalysis" IS
    'Analysis of placement by nominal code'
GO


ALTER TABLE "pears"."PlacementAnalysis"
    ADD NOT NULL FOREIGN KEY "Placement" ("PlacementID" ASC)
    REFERENCES "pears"."Placement" ("placementid")
    ON DELETE CASCADE
GO


CREATE INDEX "PlacementAnalysis_Period" ON "pears"."PlacementAnalysis"
    ( "PeriodAuthorised" )
GO


CREATE TRIGGER "PlacementAnalysisInsert" BEFORE INSERT ORDER 1 ON
"pears"."PlacementAnalysis"
REFERENCING NEW AS "new_PlacementAnalysis"
FOR each ROW
WHEN("new_PlacementAnalysis"."staffid" IS NULL)
BEGIN
  SET "new_PlacementAnalysis"."staffid" = "userstaffid"
exception
  WHEN others THEN
    SET "new_PlacementAnalysis"."staffid" = NULL
END
GO
```

```
COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."PlacementAnalysis"."PlacementAnalysisInsert" IS
{CREATE TRIGGER PlacementAnalysisInsert
BEFORE INSERT ORDER 1 ON
pears.PlacementAnalysis
REFERENCING NEW AS new_PlacementAnalysis
FOR each ROW
WHEN(new_PlacementAnalysis.staffid IS NULL)
BEGIN
  SET new_PlacementAnalysis.staffid=userstaffid
exception
  WHEN others THEN
    SET new_PlacementAnalysis.staffid=NULL
END
}
GO


CREATE TRIGGER "PlacementAnalysisAuthorised" BEFORE UPDATE OF
"CurrentlyAuthorised"
ORDER 1 ON "pears"."PlacementAnalysis"
REFERENCING NEW AS "NewPlaceAnal"
FOR each ROW
WHEN("NewPlaceAnal"."CurrentlyAuthorised" = 1)
BEGIN
  SET "NewPlaceAnal"."WhenAuthorised" = CURRENT TIMESTAMP;
  SET "NewPlaceAnal"."staffid" = "userstaffid"
END
GO


COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."PlacementAnalysis"."PlacementAnalysisAuthorised" IS
{CREATE TRIGGER PlacementAnalysisAuthorised
 BEFORE UPDATE OF CurrentlyAuthorised
ORDER 1 ON pears.PlacementAnalysis
REFERENCING NEW AS NewPlaceAnal
FOR each ROW
WHEN(NewPlaceAnal.CurrentlyAuthorised = 1)
BEGIN
  SET NewPlaceAnal.WhenAuthorised=CURRENT TIMESTAMP;
  SET NewPlaceAnal.staffid=userstaffid;
END
}
GO
```

```
CREATE TRIGGER "placementanalysisauditupdate" BEFORE UPDATE OF
"NominalCode",
"Percentage" ORDER 2 ON "pears"."PlacementAnalysis"
REFERENCING OLD AS "old_pa" NEW AS "new_pa"
FOR each ROW
WHEN(EXISTS(SELECT * FROM "AuditItems" WHERE "AreaName" = 'Placement' AND
"AuditFlag" = 1 AND "auditemid" = 'XXMAN0000000000015'))
BEGIN
  DECLARE "OurRef" CHAR(20);
  IF(SELECT "temp" FROM "placement" WHERE "placementid" =
"old_pa"."placementid") = 1 THEN
    SELECT "refcode" INTO "ourref" FROM "placement" WHERE "placementid" =
"old_pa"."placementid";
    IF UPDATE("NominalCode") THEN CALL
"AuditLog"('PLACEMENT',"new_pa"."placementid","string"('NominalCode Updated
Our Ref. - ',"ourref"),"old_pa"."NominalCode","new_pa"."NominalCode")
    END IF;
    IF UPDATE("Percentage") AND "round"("old_pa"."Percentage",2) <>
"round"("new_pa"."Percentage",2) THEN CALL
"AuditLog"('PLACEMENT',"new_pa"."placementid","string"('Percentage Updated
Our Ref. -
',"ourref"),"round"("old_pa"."Percentage",2),"round"("new_pa"."Percentage",2
))
    END IF
  END IF
END
GO


COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."PlacementAnalysis"."placementanalysisauditupdate" IS
{CREATE TRIGGER placementanalysisauditupdate
 BEFORE UPDATE OF NominalCode,
Percentage ORDER 2 ON pears.PlacementAnalysis
REFERENCING OLD AS old_pa NEW AS new_pa
FOR each ROW
WHEN(EXISTS(SELECT * FROM AuditItems WHERE AreaName = 'Placement' AND
AuditFlag = 1 AND auditemid = 'XXMAN0000000000015'))
BEGIN
  DECLARE OurRef CHAR(20);
  IF(SELECT temp FROM placement WHERE placementid = old_pa.placementid) = 1
THEN
    SELECT refcode INTO ourref FROM placement WHERE placementid =
old_pa.placementid;
    IF UPDATE(NominalCode) THEN CALL
AuditLog('PLACEMENT',new_pa.placementid,string('NominalCode Updated Our Ref.
- ',ourref),old_pa.NominalCode,new_pa.NominalCode)
```

```
    END IF;
    IF UPDATE(Percentage) AND round(old_pa.Percentage,2) <>
round(new_pa.Percentage,2) THEN CALL
AuditLog('PLACEMENT',new_pa.placementid,string('Percentage Updated Our Ref.
- ',ourref),round(old_pa.Percentage,2),round(new_pa.Percentage,2))
    END IF
  END IF
END
}
GO


CREATE TRIGGER "placementanalysisauditinsdel" BEFORE INSERT,DELETE ORDER 1
ON
"pears"."PlacementAnalysis"
REFERENCING OLD AS "old_pa" NEW AS "new_pa"
FOR each ROW
WHEN(EXISTS(SELECT * FROM "AuditItems" WHERE "AreaName" = 'Placement' AND
"AuditFlag" = 1 AND "auditemid" = 'XXMAN0000000000015'))
BEGIN
  DECLARE "OurRef" CHAR(20);
  IF inserting THEN
    IF(SELECT "temp" FROM "placement" WHERE "placementid" =
"new_pa"."placementid") = 1 THEN
      SELECT "refcode" INTO "ourref" FROM "placement" WHERE "placementid" =
"new_pa"."placementid";
      CALL "AuditLog"('PLACEMENT',"new_pa"."placementid","string"('Split
Inserted Our Ref. - ',"ourref"),'',"new_pa"."NominalCode" || ' ' ||
"round"("new_pa"."Percentage",2))
    END IF END IF;
  IF deleting THEN
    IF(SELECT "temp" FROM "placement" WHERE "placementid" =
"old_pa"."placementid") = 1 THEN
      SELECT "refcode" INTO "ourref" FROM "placement" WHERE "placementid" =
"old_pa"."placementid";
      CALL "AuditLog"('PLACEMENT',"old_pa"."placementid","string"('Split
Deleted Our Ref. - ',"ourref"),"old_pa"."NominalCode" || ' ' ||
"round"("old_pa"."Percentage",2),'')
    END IF
  END IF
END
GO


COMMENT TO PRESERVE FORMAT ON TRIGGER
"pears"."PlacementAnalysis"."placementanalysisauditinsdel" IS
{CREATE TRIGGER placementanalysisauditinsdel
 BEFORE INSERT,DELETE ORDER 1 ON
```

```
pears.PlacementAnalysis
REFERENCING OLD AS old_pa NEW AS new_pa
FOR each ROW
WHEN(EXISTS(SELECT * FROM AuditItems WHERE AreaName = 'Placement' AND
AuditFlag = 1 AND auditemid = 'XXMAN0000000000015'))
BEGIN
  DECLARE OurRef CHAR(20);
  IF inserting THEN
    IF(SELECT temp FROM placement WHERE placementid = new_pa.placementid) =
1 THEN
      SELECT refcode INTO ourref FROM placement WHERE placementid =
new_pa.placementid;
      CALL AuditLog('PLACEMENT',new_pa.placementid,string('Split Inserted
Our Ref. - ',ourref),'',new_pa.NominalCode || ' ' ||
round(new_pa.Percentage,2))
    END IF
  END IF;
  IF deleting THEN
    IF(SELECT temp FROM placement WHERE placementid = old_pa.placementid) =
1 THEN
      SELECT refcode INTO ourref FROM placement WHERE placementid =
old_pa.placementid;
      CALL AuditLog('PLACEMENT',old_pa.placementid,string('Split Deleted Our
Ref. - ',ourref),old_pa.NominalCode || ' ' || round(old_pa.Percentage,2),'')
    END IF
  END IF
END
}
GO
```