



pears.IQXPopulateAccountStatisticsIQAc



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

Original SQL

```
CREATE PROCEDURE "pears"."IQXPopulateAccountStatisticsIQAc"( IN @FromDate
DATE,IN @ToDate DATE,IN @TransferBatch INTEGER,IN @AccountFilter CHAR(30),IN
@DivFilter CHAR(30),IN @DeptFilter CHAR(30) )
BEGIN
--
--
--
DECLARE @AnalysisBatch INTEGER;
DECLARE @DoNothing SMALLINT;
DECLARE @AccountFilterLike CHAR(40);
DECLARE @DivFilterLike CHAR(40);
DECLARE @DeptFilterLike CHAR(40);
DECLARE @TimeSheetLines INTEGER;
--
--
-- create temporary table (DO NOT ALTER THIS SECTION)
--
--
--
-- work out the like clause if any
--
--
IF(@AccountFilter IS NULL OR "trim"(@AccountFilter) = '') THEN
SET @AccountFilterLike = '%'
ELSE
SET @AccountFilterLike = "string"('[',"trim"(@AccountFilter),']%')
END IF;
IF(@DivFilter IS NULL OR "trim"(@DivFilter) = '') THEN
SET @DivFilterLike = '%'
ELSE
SET @DivFilterLike = "string"('[',"trim"(@DivFilter),']%')
END IF;
IF(@DeptFilter IS NULL OR "trim"(@DeptFilter) = '') THEN
SET @DeptFilterLike = '%'
```



```
ELSE
    SET @DeptFilterLike = "string"('[', "trim"(@DeptFilter), ']%')
END IF;
--
--
-- Are there any Timesheet lines that cover the search
criocumentdateteria, if not abort
--
--
IF "IsNull"(@TransferBatch,0) > 0 THEN
    SELECT "count"()
        INTO @TimeSheetLines
        FROM "temptimesheetline" KEY JOIN "temptimesheet" AS "t" LEFT OUTER
JOIN("placement" KEY JOIN "employment" KEY JOIN "company")
        WHERE "isnull"("t"."TransferBatch",0) > 0 AND "company"."clientcode"
LIKE @AccountFilterLike
        AND "isnull"("t"."AnalysisBatch",0) = 0
        AND "isnull"("company"."divisionid", '') LIKE @DivFilterLike
        AND "isnull"("placement"."departmentid", '') LIKE @DeptFilterLike
ELSE
    SELECT "count"()
        INTO @TimeSheetLines
        FROM "temptimesheetline" KEY JOIN "temptimesheet" AS "t" LEFT OUTER
JOIN("placement" KEY JOIN "employment" KEY JOIN "company")
        WHERE "company"."clientcode" LIKE @AccountFilterLike AND
"isnull"("t"."AnalysisBatch",0) = 0
        AND "date"("t"."WhenEntered") BETWEEN @FromDate AND @ToDate
        AND "isnull"("company"."divisionid", '') LIKE @DivFilterLike
        AND "isnull"("placement"."departmentid", '') LIKE @DeptFilterLike
END IF;
IF "isnull"(@TimeSheetLines,0) = 0 THEN RETURN END IF;
--
--
-- Get the next analysis batch number
--
--
BEGIN
    UPDATE "params" SET "LastAnalysisBatch" =
"isnull"("LastAnalysisBatch",0)+1;
    SELECT "LastAnalysisBatch" INTO @AnalysisBatch FROM "params";
    commit WORK
END;
-- Should only be needed if debugging - table should not exist
BEGIN
    DROP TABLE "pears"."TEMPAccountsStatistics"
exception
    WHEN others THEN SET @DoNothing = 1
END;
```



```
--
-- create temporary table (DO NOT ALTER THIS SECTION)
--
CREATE LOCAL TEMPORARY TABLE "pears"."TEMPAccountsStatistics" (
  "AccountsStatisticsID" CHAR(20) NOT NULL,
  "AccountingDate" DATE NULL,
  "Amount" NUMERIC(12,2) NULL,
  "Description" CHAR(100) NULL,
  "Type" CHAR(20) NULL,
  "NominalCodeSegment1" CHAR(12) NULL,
  "NominalCodeSegment2" CHAR(12) NULL,
  "NominalCodeSegment3" CHAR(12) NULL,
  "NominalCodeSegment4" CHAR(12) NULL,
  "NominalCodeSegment5" CHAR(12) NULL,
  "PayrollCompany" CHAR(10) NULL,
  "DivisionAnalysisCode" CHAR(50) NULL,
  "DepartmentAnalysisCode" CHAR(50) NULL,
  "TimesheetAnalysisCode" CHAR(50) NULL,
  "TimesheetConsultantAnalysisCode" CHAR(50) NULL,
  "PlacementConsultantAnalysisCode" CHAR(50) NULL,
  "vacancyConsultantAnalysisCode" CHAR(50) NULL,
  "ConsultantAnalysisCode" CHAR(50) NULL,
  "ShiftConsultantAnalysisCode" CHAR(50) NULL,
  "ClientAccountCode" CHAR(20) NULL,
  "CompanyID" CHAR(20) NULL,
  "PersonID" CHAR(20) NULL,
  "TempTimesheetID" CHAR(20) NULL,
  "TempTimesheetLineID" CHAR(20) NULL,
  "TempDeskID" CHAR(20) NULL,
  "TimeSheetStaffID" CHAR(20) NULL,
  "VacancyStaffID" CHAR(20) NULL,
  "PlacementStaffID" CHAR(20) NULL,
  "PayrollNumber" CHAR(20) NULL,
  "PersonName" CHAR(50) NULL,
  "CompanyName" CHAR(50) NULL,
  "TimesheetSerial" CHAR(20) NULL,
  "TimesheetLineNumber" SMALLINT NULL,
  "DocumentNumber" CHAR(20) NULL,
  "DocumentID" CHAR(20) NULL,
  "DocumentDate" DATE NULL,
  "JournalLineNumber" SMALLINT NULL,
  "SalesOrCost" CHAR(1) NULL,
  "Currency" CHAR(3) NULL,
  "PlacementID" CHAR(20) NULL,
  "VacancyID" CHAR(20) NULL,
  "TempShiftID" CHAR(20) NULL,
  "TempShiftDate" DATE NULL,
  "DocumentLineType" CHAR(1) NULL,
```



```
"Quantity" DOUBLE NULL,  
"JournalClass" CHAR(20) NULL,  
"Unitdescription" CHAR(50) NULL,  
"PayBandPayrollFlag" CHAR(10) NULL, // payrollflag  
"IsExpenses" SMALLINT NULL,  
"TaxMethod" INTEGER NULL,  
"Paid" DOUBLE NULL,  
"HolPay" DOUBLE NULL,  
"HolLiab" DOUBLE NULL,  
"AWRHolRate" DOUBLE NULL,  
"Levy" DOUBLE NULL,  
"HolLiabNI" DOUBLE NULL,  
"ActualNI" DOUBLE NULL,  
"Pension" DOUBLE NULL,  
"Statutory" DOUBLE NULL,  
"Charged" DOUBLE NULL,  
"TempdeskAnalysiscode" CHAR(20) NULL,  
"GoodsAmount" NUMERIC(12,2) NULL,  
"PriceEach" DOUBLE NULL,  
"VatAmount" DOUBLE NULL,  
"VatCode" CHAR(3) NULL,  
"JournalNominalCode" CHAR(12) NULL,  
"PayrollIdentifier" CHAR(1) NULL,  
"Period" INTEGER NULL,  
"TimesheetSupplierCode" CHAR(12) NULL,  
"ShiftAnalysisCode" CHAR(20) NULL,  
"ShiftReferenceCode" CHAR(20) NULL,  
"ShiftSerialNumber" BIGINT NULL,  
"TempShiftTypeID" CHAR(20) NULL,  
"AnalysisBatch" INTEGER NULL,  
"TransferBatch" INTEGER NULL,  
"EmploymentPosition" CHAR(50) NULL,  
"EssentialSkillGradeID" CHAR(4) NULL,  
"vacancytheirref" CHAR(50) NULL,  
"vacancyrefcode" CHAR(20) NULL,  
"CustomString1" CHAR(40) NULL,  
"CustomString2" CHAR(40) NULL,  
"CustomString3" CHAR(40) NULL,  
"CustomString4" CHAR(40) NULL,  
"CustomString5" CHAR(40) NULL,  
"CustomDouble1" DOUBLE NULL,  
"CustomDouble2" DOUBLE NULL,  
"CustomDouble3" DOUBLE NULL,  
"CustomDouble4" DOUBLE NULL,  
"CustomDouble5" DOUBLE NULL,  
PRIMARY KEY("AccountsStatisticsID"),  
) ON commit DELETE ROWS;  
CREATE INDEX "a" ON "pears"."TEMPAccountsStatistics" ("TempTimesheetID");
```



```
CREATE INDEX "b" ON "pears"."TEMPAccountsStatistics" ("PersonID");
CREATE INDEX "c" ON "pears"."TEMPAccountsStatistics" ("PayrollNumber");
CREATE INDEX "d" ON
"pears"."TEMPAccountsStatistics" ("TempTimesheetLineID");
CREATE INDEX "e" ON
"pears"."TEMPAccountsStatistics" ("NominalCodeSegment1");
CREATE INDEX "f" ON
"pears"."TEMPAccountsStatistics" ("NominalCodeSegment2");
CREATE INDEX "g" ON
"pears"."TEMPAccountsStatistics" ("NominalCodeSegment3");
CREATE INDEX "h" ON
"pears"."TEMPAccountsStatistics" ("NominalCodeSegment4");
CREATE INDEX "i" ON
"pears"."TEMPAccountsStatistics" ("NominalCodeSegment5");
CREATE INDEX "j" ON "pears"."TEMPAccountsStatistics" ("PlacementID");
CREATE INDEX "l" ON
"pears"."TEMPAccountsStatistics" ("TempTimesheetLineID");
CREATE INDEX "k" ON "pears"."TEMPAccountsStatistics" ("DocumentID");
BEGIN atomic
  --
  --
  -- Update the Temptimesheet lines with the new Analysis batch
  -- if transfer batch > 0 then use this as a filter, ie records in
transfer batch will mirror the transfer batch number
  -- if transferbatch = 0, then pick up journal records in the date range
@PeriodFrom to @periodto
  --
  IF "Isnull"(@TransferBatch,0) > 0 THEN
    UPDATE "Temptimesheet" AS "t"
      SET "t"."AnalysisBatch" = @AnalysisBatch FROM
        "temptimesheet" AS "t" LEFT OUTER JOIN("placement" KEY JOIN
"employment" KEY JOIN "company")
        WHERE "isnull"("t"."TransferBatch",0) = @transferbatch AND
"company"."clientcode" LIKE @AccountFilterLike
        AND "isnull"("t"."AnalysisBatch",0) = 0
        AND "isnull"("company"."divisionid",'') LIKE @DivFilterLike
        AND "isnull"("placement"."departmentid",'') LIKE @DeptFilterLike
  ELSE
    UPDATE "Temptimesheet" AS "t"
      SET "t"."AnalysisBatch" = @AnalysisBatch FROM
        "temptimesheet" AS "t" LEFT OUTER JOIN("placement" KEY JOIN
"employment" KEY JOIN "company")
        WHERE "company"."clientcode" LIKE @AccountFilterLike AND
"isnull"("t"."AnalysisBatch",0) = 0
        AND "date"("t"."whenentered") BETWEEN @FromDate AND @ToDate
        AND "isnull"("company"."divisionid",'') LIKE @DivFilterLike
        AND "isnull"("placement"."departmentid",'') LIKE @DeptFilterLike
  END IF;
```



```
--
--
--  Populate temporary table with Data, fixed data first. Should not
needs to change this section
--
--
--
--
--  timesheet line related details, dont change this section
--
--
INSERT INTO "TEMPAccountsStatistics"( "AccountsStatisticsID",
"TemptimesheetID","TimesheetSerial","TimesheetLineNumber","Personid","Payrol
lNumber","Placementid","Tempdeskid",
"TimesheetAnalysisCode","tempshiftid","paid","PayBandPayRollFlag","IsExpense
s","charged","AnalysisBatch",
"TaxMethod","TempTimesheetLineID","Period","Quantity",
"Unitdescription","accountingdate","transferbatch" )
SELECT "uniquekey"("number"()),
"tl"."TempTimesheetID",
"t"."SerialNumber",
"tl"."LineNumber",
"t"."personid",
"t"."payrollnumber",
"t"."PlacementID",
"t"."TempDeskID",
"t"."AnalysisCode",
"tl"."TempShiftID",
"isnull"("tl"."unitspaid"*"tl"."payrate",0) AS "paid",
"PayrollFlag",
"ISNULL"("IsExpenses",0),
"isnull"("tl"."UnitsCharged"*"tl"."ChargeRate",0) AS "charged",
"AnalysisBatch","TaxMethod","tl"."TempTimesheetLineID","Period",
"isnull"("unitspaid",0),
"b"."unit","whenentered","t"."transferbatch"
FROM "TempTimesheetLine" AS "tl"
KEY JOIN("TempTimesheet" AS "t","temppayband" AS "b") WHERE
"analysisbatch" = @AnalysisBatch;
--
--
UPDATE "TEMPAccountsStatistics" AS "tas" SET "journallinenum" =
"j"."linenum","amount" = "j"."amount",
"goodsamount" = "j"."goodsamount","JournalNominalCode" =
"j"."NominalCode",
"journalclass" = "j"."journalclass","pric each" =
"j"."pric each","vatamount" = "j"."vatamount","vatcode" = "j"."vatcode",
"description" = "j"."description","documentid" = "j"."documentid" FROM
"TEMPAccountsStatistics" AS "tas" JOIN "iqacjournal" AS "j" ON "xref"
```



```
= 'T' AND "xrefid" = "tas"."temptimesheetlineid";
    UPDATE "TEMPAccountsStatistics" AS "tas" SET "documentnumber" =
"ourref","DocumentDate" = "d"."Documentdate" FROM
    "TEMPAccountsStatistics" AS "tas" JOIN "iqacddocument" AS "d" ON
"tas"."documentid" = "d"."documentid" AND "d"."ledgerid" = 'SALES';
--
-- Person ID when no timesheet line
--
--
    UPDATE "TEMPAccountsStatistics" AS "tas"
    SET "vacancyid" = "v"."vacancyid","vacancytheirref" =
"v"."theirref","vacancyrefcode" = "v"."refcode" FROM
    "TEMPAccountsStatistics" AS "tas" JOIN "placement" AS "p" ON
"tas"."placementID" = "p"."placementID" KEY JOIN ("vacancy" AS
"v","employment" AS "e");
--
--
-- Employment Position, Placement Temp, Currency
--
--
    UPDATE "TEMPAccountsStatistics" AS "tas" SET "EmploymentPosition" =
"Employment"."Position",
    "currency" = "isnull"("tas"."currency","p"."currency") FROM
    "TEMPAccountsStatistics" AS "tas" JOIN "placement" AS "p" ON
"tas"."placementID" = "p"."placementID"
    ,"placement" AS "p" KEY JOIN "employment";
--
-- Employment Position, Placement Temp, Currency
--
--
    UPDATE "TEMPAccountsStatistics" AS "tas"
    SET "Period" = "t"."Period","payrollidentifier" =
"t"."payrollidentifier","TimesheetSupplierCode" = "t"."SupplierCode" FROM
    "TEMPAccountsStatistics" AS "tas" JOIN "TempTimesheet" AS "t" ON
"tas"."TempTimesheetID" = "t"."TempTimesheetID";
--
-- Shift Fields
--
--
    UPDATE "TEMPAccountsStatistics" AS "tas"
    SET "ShiftReferenceCode" = "t"."referencecode","ShiftSerialNumber" =
"p"."shiftserialnumber","tempshiftdate" = "t"."shiftdate",
    "Shiftanalysiscode" = "t"."AnalysisCode","EssentialSkillGradeID" =
"t"."EssentialSkillGradeID",
    "TempShifttypeid" =
"t"."TempShifttypeid","shiftconsultantAnalysisCode" = (SELECT
"s"."analysiscode" FROM "staff" AS "s" WHERE "s"."staffid" = "t"."staffid")
FROM
```



```
"TEMPAccountsStatistics" AS "tas" JOIN "TempShift" AS "t" ON
"tas"."TempShiftID" = "t"."TempShiftID"
, "TempShift" AS "t" KEY JOIN "TempShiftplan" AS "p";
--
--
-- Person names
--
--
UPDATE "TEMPAccountsStatistics" AS "tas" SET "personname" = "p"."name"
FROM "TEMPAccountsStatistics" AS "tas" JOIN "person" AS "p" ON
"tas"."personid" = "p"."personid";
--
--
-- Company Details for placement related invoices
--
--
UPDATE "TEMPAccountsStatistics" AS "tas"
SET "CompanyID" = "c"."CompanyID",
"CompanyName" = "c"."Name",
"ClientAccountCode" = "c"."clientcode" FROM
"TEMPAccountsStatistics" AS "tas" JOIN "placement" AS "p" ON
"tas"."placementID" = "p"."placementID"
KEY JOIN "vacancy" KEY JOIN "employment" AS "e" KEY JOIN "company" AS
"c";
--
--
--
-- Placement Nominal Codes
--
--
UPDATE "TEMPAccountsStatistics" AS "tas" SET "NominalCodeSegment1" =
(SELECT FIRST "nominalcode"
FROM (SELECT top 1 "nominalcode" FROM "placementanalysis" AS "pl1"
, (SELECT "count"() AS "plc" FROM "placementanalysis" AS "pl"
WHERE "pl"."placementid" = "tas"."placementid") AS "t2"
WHERE "pl1"."placementid" = "tas"."placementid" AND "plc" >
ORDER BY "whenauthorised" ASC) AS "t1" ORDER BY "nominalcode" DESC) WHERE
"placementid" IS NOT NULL;
UPDATE "TEMPAccountsStatistics" AS "tas"
SET "NominalCodeSegment2"
= (SELECT FIRST "nominalcode"
FROM (SELECT top 2 "nominalcode" FROM "placementanalysis" AS "pl1"
, (SELECT "count"() AS "plc" FROM "placementanalysis" AS "pl"
WHERE "pl"."placementid" = "tas"."placementid") AS "t2"
WHERE "pl1"."placementid" = "tas"."placementid" AND "plc" > 1
ORDER BY "whenauthorised" ASC) AS "t1" ORDER BY "nominalcode" DESC) WHERE
"placementid" IS NOT NULL AND "NominalCodeSegment1" IS NOT NULL;
UPDATE "TEMPAccountsStatistics" AS "tas" SET "NominalCodeSegment3" =
```



```
(SELECT FIRST "nominalcode"
  FROM(SELECT top 3 "nominalcode" FROM "placementanalysis" AS "pl1"
    ,(SELECT "count"() AS "plc" FROM "placementanalysis" AS "pl"
      WHERE "pl"."placementid" = "tas"."placementid") AS "t2"
    WHERE "pl1"."placementid" = "tas"."placementid" AND "plc" > 2
  ORDER BY "whenauthorised" ASC) AS "t1" ORDER BY "nominalcode" DESC) WHERE
"placementid" IS NOT NULL AND "NominalCodeSegment2" IS NOT NULL;
  UPDATE "TEMPAccountsStatistics" AS "tas" SET "NominalCodeSegment4" =
(SELECT FIRST "nominalcode"
  FROM(SELECT top 4 "nominalcode" FROM "placementanalysis" AS "pl1"
    ,(SELECT "count"() AS "plc" FROM "placementanalysis" AS "pl"
      WHERE "pl"."placementid" = "tas"."placementid") AS "t2"
    WHERE "pl1"."placementid" = "tas"."placementid" AND "plc" > 3
  ORDER BY "whenauthorised" ASC) AS "t1" ORDER BY "nominalcode" DESC) WHERE
"placementid" IS NOT NULL AND "NominalCodeSegment3" IS NOT NULL;
  UPDATE "TEMPAccountsStatistics" AS "tas" SET "NominalCodeSegment5" =
(SELECT FIRST "nominalcode"
  FROM(SELECT top 5 "nominalcode" FROM "placementanalysis" AS "pl1"
    ,(SELECT "count"() AS "plc" FROM "placementanalysis" AS "pl"
      WHERE "pl"."placementid" = "tas"."placementid") AS "t2"
    WHERE "pl1"."placementid" = "tas"."placementid" AND "plc" > 4
  ORDER BY "whenauthorised" ASC) AS "t1" ORDER BY "nominalcode" DESC) WHERE
"placementid" IS NOT NULL AND "NominalCodeSegment4" IS NOT NULL;
  --
  --
  -- StaffID
  --
  --
  UPDATE "TEMPAccountsStatistics" AS "tas" SET "TimesheetStaffID" =
(SELECT "staffid" FROM "temptimesheet" AS "t" WHERE "t"."temptimesheetid" =
"tas"."temptimesheetid"),
  "PlacementStaffID" = (SELECT "staffid" FROM "placement" AS "p" WHERE
"p"."placementid" = "tas"."placementid"),
  "vacancyStaffID" = (SELECT "staffid" FROM "vacancy" AS "v" WHERE
"v"."vacancyid" = "tas"."vacancyid");
  --
  --
  -- Analysis Codes
  --
  --
  UPDATE "TEMPAccountsStatistics" AS "tas"
  SET "DivisionAnalysisCode" = "isnull"((SELECT "Analysis" FROM
"Division" KEY JOIN "TempDesk" WHERE "TempDesk"."TempDeskID" =
"tas"."TempDeskID"),
  (SELECT "Analysis" FROM "Division" KEY JOIN "Company" WHERE
"tas"."CompanyID" = "Company"."CompanyID")),
  "DepartmentAnalysisCode" = (SELECT "Analysis" FROM "Department" KEY
JOIN "Placement" WHERE "Placement"."PlacementID" = "tas"."PlacementID"),
```



```
"TempdeskAnalysiscode" = "isnull"((SELECT "Defanalysiscode" FROM
"TempDesk" WHERE "TempDesk"."TempDeskID" = "tas"."TempDeskID")),'),
"VacancyConsultantAnalysisCode" = (SELECT "s"."analysiscode" FROM
"staff" AS "s" WHERE "tas"."vacancystaffid" = "s"."staffid"),
"PlacementConsultantAnalysisCode" = (SELECT "s"."analysiscode" FROM
"staff" AS "s" WHERE "tas"."placementstaffid" = "s"."staffid"),
"TimeSheetConsultantAnalysisCode" = (SELECT "s"."analysiscode" FROM
"staff" AS "s" WHERE "tas"."timesheetstaffid" = "s"."staffid"),
"ConsultantAnalysisCode" = (SELECT "s"."analysiscode" FROM "person"
KEY JOIN "staff" AS "s" WHERE "tas"."personid" = "person"."personid");
UPDATE "TEMPAccountsStatistics" AS "tas" SET "personname" = "p"."name"
FROM "TEMPAccountsStatistics" AS "tas" JOIN "person" AS "p" ON
"tas"."personid" = "p"."personid";
--
--
-- Now the extras which WILL different per site
--
--un
--
-- Pensions, Holiday, Levy
--
UPDATE "TEMPAccountsStatistics" AS "tas"
SET "HolPay" = IF "IncludeInHolidayPay" = '1' THEN "paid" ELSE 0 endif
FROM
"TEMPAccountsStatistics" AS "tas" JOIN "temptimesheetline" AS "l"
ON("l"."temptimesheetlineid" =
"tas"."temptimesheetlineid"), "temptimesheetline" AS "l" KEY JOIN
"temppayband" WHERE "tas"."temptimesheetlineid"
= "l"."temptimesheetlineid";
UPDATE "TEMPAccountsStatistics" AS "tas"
SET "AWRHolRate"
= IF "isnull"("AWRWasInvolved",0) = 0 THEN
"isnull"((SELECT FIRST("ExtraHols"/("52-"ExtraHols")) FROM
"AWRJobMaster" AS "a" WHERE "a"."placementid" = "tas"."placementid"),0) ELSE
0 endif FROM
"TEMPAccountsStatistics" AS "tas" JOIN "temptimesheet" AS "t" ON
"tas"."temptimesheetid" = "t"."temptimesheetid";
-- update "TEMPAccountsStatistics" as "tas" set "HolLiab" =
"PayRate"*"UnitsPaid" from
-- "temptimesheetline" key join "temppayband" as "b" where
"b"."description" like '%HP';
-- update "TEMPAccountsStatistics" as "tas" set "Levy" =
"isnull"((select "sum"("amount") from "accord"."IQXJournalHistory" where
"trim"("cosegmenta") = "timesheetserial" and("wtcode" = '603950')),0);
-- update "TEMPAccountsStatistics" as "tas" set "HolLiabNI" =
"iqmoneyround"("HolLiab"*.1);
-- update "TEMPAccountsStatistics" as "tas" set "ActualNI" =
"isnull"((select "sum"("amount") from "accord"."IQXJournalHistory" where
```



```
"trim"("cosegmenta") = "timesheetserial" and(("wtcode" = '603000')
or("wtcode" = '603500'))),0);
-- update "TEMPAccountsStatistics" as "tas" set "Pension" =
"isnull"((select "sum"("amount") from "accord"."iQXJournalHistory" where
"trim"("cosegmenta") = "timesheetserial" and("wtcode" = '540620'))),0);
UPDATE "TEMPAccountsStatistics" AS "tas" SET "Statutory" = 0;
UPDATE "TEMPAccountsStatistics" AS "tas" SET "Charged" =
"Charged"+"Pension"
WHERE EXISTS(SELECT * FROM "placement" AS "p" KEY JOIN "employment"
KEY JOIN "companyaccount" WHERE "p"."placementid" = "tas"."placementid" AND
"ERNIoninvoice" = 1)
AND EXISTS(SELECT * FROM "temptimesheetline" AS "l" WHERE
"l"."temptimesheetlineid" = "tas"."temptimesheetlineid" AND "l"."linenumber"
= 1);
--
-- questions etc, WILL different per site
--
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomDouble1" =
(SELECT("unitscharged"*"chargerate")-(("unitspaid"*"payrate") FROM
"temptimesheetline" AS "t" WHERE "tas"."temptimesheetlineid" =
"t"."temptimesheetlineid");
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomDouble2" =
(("paid"/"charged")*100) WHERE "charged" <> 0;
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomDouble3" = "paid";
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomDouble4" =
"charged";
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomDouble5" = 0;
UPDATE "TEMPAccountsStatistics" SET "type" = '';
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomString1" = (SELECT
"department"."name" FROM "placement" KEY JOIN "department" WHERE
"placement"."placementid" = "tas"."placementid");
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomString2" = (SELECT
"staff"."userid" FROM "staff" WHERE "staff"."staffid" =
"tas"."vacancystaffid");
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomString3" = (SELECT
"keyname" FROM "person" WHERE "person"."personid" = "tas"."personid");
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomString4" = (SELECT
"surname" FROM "person" WHERE "person"."personid" = "tas"."personid");
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomString5" = (SELECT
"forenames" FROM "person" WHERE "person"."personid" = "tas"."personid");
--
--
-- Finally Insert the temp table into the Main Accounts Stats table - Do
not modify
--
--
--
--
```



```
-- delete from "TEMPAccountsStatistics";
--
--
-- Insert the Analysis Batch into the batch transfer table- Do not
modify
INSERT INTO "IQXAccountsStatistics"
(
"IXAccountsStatisticsID", "AccountingDate", "Amount", "Description", "Type",
"NominalCodeSegment1", "NominalCodeSegment2", "NominalCodeSegment3", "NominalCo
deSegment4", "NominalCodeSegment5", "PayrollCompany",
"DivisionAnalysisCode", "DepartmentAnalysisCode", "TimesheetAnalysisCode", "Tim
esheetConsultantAnalysisCode", "VacancyConsultantAnalysisCode", "PlacementCons
ultantAnalysisCode", "ConsultantAnalysisCode",
"ClientAccountCode", "CompanyID", "PersonID",
"TempTimesheetID", "TempTimesheetLineID", "TempDeskID", "VacancyStaffID", "Place
mentStaffid", "TimeSheetStaffid", "PayrollNumber", "PersonName", "CompanyName", "
TimesheetSerial",
"TimesheetLineNumber", "DocumentNumber", "DocumentID", "DocumentDate", "JournalL
ineNumber", "SalesOrCost", "Currency", "PlacementID", "VacancyID", "TempShiftID",
"tempshiftdate", "DocumentLineType",
"JournalClass", "Quantity", "Unitdescription", "TaxMethod", "PaID", "HolPay", "Hol
Liab", "AWRHolRate", "Levy",
"HolLiabNI", "ActualNI", "Pension", "Statutory", "Charged", "TempdeskAnalysiscode
", "GoodsAmount", "priceEach", "VatAmount", "VatCode", "JournalNominalCode",
"PayrollIdentifier", "Period", "TimesheetSupplierCode", "ShiftAnalysisCode", "Sh
iftReferenceCode", "ShiftSerialNumber", "TempShiftTypeID",
"CustomString1", "CustomString2", "CustomString3", "CustomString4", "CustomStrin
g5", "CustomDouble1", "CustomDouble2", "CustomDouble3",
"CustomDouble4", "CustomDouble5", "AnalysisBatch", "EmploymentPosition", "Essent
ialSkillGradeID", "PayBandPayrollFlag", "IsExpenses", "VacancyTheirRef", "Vacanc
yRefcode", "ShiftConsultantAnalysisCode",
"transferbatch" )
SELECT
"AccountsStatisticsID", "AccountingDate", "Amount", "Description", "Type",
"NominalCodeSegment1", "NominalCodeSegment2", "NominalCodeSegment3", "NominalCo
deSegment4", "NominalCodeSegment5", "PayrollCompany",
"DivisionAnalysisCode", "DepartmentAnalysisCode", "TimesheetAnalysisCode", "Tim
esheetConsultantAnalysisCode", "VacancyConsultantAnalysisCode", "PlacementCons
ultantAnalysisCode", "ConsultantAnalysisCode",
"ClientAccountCode", "CompanyID", "PersonID",
"TempTimesheetID", "TempTimesheetLineID", "TempDeskID", "VacancyStaffID", "Place
mentStaffid", "TimeSheetStaffid", "PayrollNumber", "PersonName", "CompanyName", "
TimesheetSerial",
"TimesheetLineNumber", "DocumentNumber", "DocumentID", "DocumentDate", "JournalL
ineNumber", "SalesOrCost", "Currency", "PlacementID", "VacancyID", "TempShiftID",
"tempshiftdate", "DocumentLineType",
"JournalClass", "Quantity", "Unitdescription", "TaxMethod", "PaID", "HolPay", "Hol
Liab", "AWRHolRate", "Levy",
```



```
"HolLiabNI", "ActualNI", "Pension", "Statutory", "Charged", "TempdeskAnalysiscode",
"GoodsAmount", "priceEach", "VatAmount", "VatCode", "JournalNominalCode",
"PayrollIdentifier", "Period", "TimesheetSupplierCode", "ShiftAnalysisCode", "ShiftReferenceCode", "ShiftSerialNumber", "TempShiftTypeID",
"CustomString1", "CustomString2", "CustomString3", "CustomString4", "CustomString5", "CustomDouble1", "CustomDouble2", "CustomDouble3",
"CustomDouble4", "CustomDouble5", "AnalysisBatch", "EmploymentPosition", "EssentialSkillGradeID", "PayBandPayrollFlag", "IsExpenses", "VacancyTheirRef", "VacancyRefcode", "ShiftConsultantAnalysisCode",
    "transferbatch"
FROM "TEMPAccountsStatistics";

--
INSERT INTO "BatchTransfers"(
"BatchType", "BatchNumber", "TransferTime", "Note" ) VALUES
    ( 'IQXAnalysis', @AnalysisBatch, CURRENT_TIMESTAMP, "string('Accounts
Statistics Export by ', (SELECT "userid" FROM "staff" WHERE "staffid" =
"userstaffid")) ) ;
    -- drop the temporary table
    DROP TABLE "pears"."TEMPAccountsStatistics"
    -- return the transfer batch
    -- return(@AnalysisBatch)
END // END OF atomic statement
END
GO

COMMENT TO PRESERVE FORMAT ON PROCEDURE
"pears"."IQXPopulateAccountStatisticsIQAc" IS
{CREATE PROCEDURE pears."IQXPopulateAccountStatisticsIQAc"(IN @FromDate
DATE, IN @ToDate DATE, IN @TransferBatch INTEGER, IN @AccountFilter CHAR(30),
IN @DivFilter CHAR(30), IN @DeptFilter CHAR(30) )
BEGIN
    --
    --
    --
    DECLARE @AnalysisBatch INTEGER;
    DECLARE @DoNothing SMALLINT;
    DECLARE @AccountFilterLike CHAR(40);
    DECLARE @DivFilterLike CHAR(40);
    DECLARE @DeptFilterLike CHAR(40);
    DECLARE @TimeSheetLines INTEGER;
    --
    --
    -- create temporary table (DO NOT ALTER THIS SECTION)
    --
    --
    --
    -- work out the like clause if any
```



```
--
--
IF(@AccountFilter IS NULL OR "trim"(@AccountFilter) = '') THEN
    SET @AccountFilterLike = '%'
ELSE
    SET @AccountFilterLike = "string"('[', "trim"(@AccountFilter), ']%')
END IF;
IF(@DivFilter IS NULL OR "trim"(@DivFilter) = '') THEN
    SET @DivFilterLike = '%'
ELSE
    SET @DivFilterLike = "string"('[', "trim"(@DivFilter), ']%')
END IF;
IF(@DeptFilter IS NULL OR "trim"(@DeptFilter) = '') THEN
    SET @DeptFilterLike = '%'
ELSE
    SET @DeptFilterLike = "string"('[', "trim"(@DeptFilter), ']%')
END IF;
--
--
-- Are there any Timesheet lines that cover the search
criocumentdateteria, if not abort
--
--
IF "IsNull"(@TransferBatch,0) > 0 THEN
    SELECT "count"()
        INTO @TimeSheetLines
        FROM "temptimesheetline" KEY JOIN "temptimesheet" AS "t" LEFT OUTER
JOIN("placement" KEY JOIN "employment" KEY JOIN "company")
        WHERE "isnull"("t"."TransferBatch",0) >0 AND "company"."clientcode"
LIKE @AccountFilterLike
        AND "isnull"("t"."AnalysisBatch",0) = 0
        AND "isnull"("company"."divisionid", '') LIKE @DivFilterLike
        AND "isnull"("placement"."departmentid", '') LIKE @DeptFilterLike
ELSE
    SELECT "count"()
        INTO @TimeSheetLines
        FROM "temptimesheetline" KEY JOIN "temptimesheet" AS "t" LEFT OUTER
JOIN("placement" KEY JOIN "employment" KEY JOIN "company")

        WHERE "company"."clientcode" LIKE @AccountFilterLike AND
"isnull"("t"."AnalysisBatch",0) = 0
        AND "date"("t"."WhenEntered") BETWEEN @FromDate AND @ToDate
        AND "isnull"("company"."divisionid", '') LIKE @DivFilterLike
        AND "isnull"("placement"."departmentid", '') LIKE @DeptFilterLike
END IF;
IF "isnull"(@TimeSheetLines,0) = 0 THEN RETURN END IF;
--
--
```



```
-- Get the next analysis batch number
--
--
BEGIN
    UPDATE "params" SET "LastAnalysisBatch" =
"isnull"("LastAnalysisBatch",0)+1;
    SELECT "LastAnalysisBatch" INTO @AnalysisBatch FROM "params";
    commit WORK
END;
-- Should only be needed if debugging - table should not exist
BEGIN
    DROP TABLE "pears"."TEMPAccountsStatistics"
exception
    WHEN others THEN SET @DoNothing = 1
END;
--
-- create temporary table (DO NOT ALTER THIS SECTION)
--
CREATE LOCAL TEMPORARY TABLE "pears"."TEMPAccountsStatistics"(
    "AccountsStatisticsID" CHAR(20) NOT NULL,
    "AccountingDate" DATE NULL,
    "Amount" NUMERIC(12,2) NULL,
    "Description" CHAR(100) NULL,
    "Type" CHAR(20) NULL,
    "NominalCodeSegment1" CHAR(12) NULL,
    "NominalCodeSegment2" CHAR(12) NULL,
    "NominalCodeSegment3" CHAR(12) NULL,
    "NominalCodeSegment4" CHAR(12) NULL,
    "NominalCodeSegment5" CHAR(12) NULL,
    "PayrollCompany" CHAR(10) NULL,
    "DivisionAnalysisCode" CHAR(50) NULL,
    "DepartmentAnalysisCode" CHAR(50) NULL,
    "TimesheetAnalysisCode" CHAR(50) NULL,
    "TimesheetConsultantAnalysisCode" CHAR(50) NULL,
    "PlacementConsultantAnalysisCode" CHAR(50) NULL,
    "vacancyConsultantAnalysisCode" CHAR(50) NULL,
    "ConsultantAnalysisCode" CHAR(50) NULL,
    "ShiftConsultantAnalysisCode" CHAR(50) NULL,
    "ClientAccountCode" CHAR(20) NULL,
    "CompanyID" CHAR(20) NULL,
    "PersonID" CHAR(20) NULL,
    "TempTimesheetID" CHAR(20) NULL,
    "TempTimesheetLineID" CHAR(20) NULL,
    "TempDeskID" CHAR(20) NULL,
    "TimeSheetStaffID" CHAR(20) NULL,
    "VacancyStaffID" CHAR(20) NULL,
    "PlacementStaffID" CHAR(20) NULL,
    "PayrollNumber" CHAR(20) NULL,
```



```
"PersonName" CHAR(50) NULL,  
"CompanyName" CHAR(50) NULL,  
"TimesheetSerial" CHAR(20) NULL,  
"TimesheetLineNumber" SMALLINT NULL,  
"DocumentNumber" CHAR(20) NULL,  
"DocumentID" CHAR(20) NULL,  
"DocumentDate" DATE NULL,  
"JournalLineNumber" SMALLINT NULL,  
"SalesOrCost" CHAR(1) NULL,  
"Currency" CHAR(3) NULL,  
"PlacementID" CHAR(20) NULL,  
"VacancyID" CHAR(20) NULL,  
"TempShiftID" CHAR(20) NULL,  
"TempShiftDate" DATE NULL,  
"DocumentLineType" CHAR(1) NULL,  
"Quantity" DOUBLE NULL,  
"JournalClass" CHAR(20) NULL,  
"Unitdescription" CHAR(50) NULL,  
"PayBandPayrollFlag" CHAR(10) NULL, // payrollflag  
"IsExpenses" SMALLINT NULL,  
"TaxMethod" INTEGER NULL,  
"Paid" DOUBLE NULL,  
"HolPay" DOUBLE NULL,  
"HolLiab" DOUBLE NULL,  
"AWRHolRate" DOUBLE NULL,  
"Levy" DOUBLE NULL,  
"HolLiabNI" DOUBLE NULL,  
"ActualNI" DOUBLE NULL,  
"Pension" DOUBLE NULL,  
"Statutory" DOUBLE NULL,  
"Charged" DOUBLE NULL,  
"TempdeskAnalysiscode" CHAR(20) NULL,  
"GoodsAmount" NUMERIC(12,2) NULL,  
"PriceEach" DOUBLE NULL,  
"VatAmount" DOUBLE NULL,  
"VatCode" CHAR(3) NULL,  
"JournalNominalCode" CHAR(12) NULL,  
"PayrollIdentifier" CHAR(1) NULL,  
"Period" INTEGER NULL,  
"TimesheetSupplierCode" CHAR(12) NULL,  
"ShiftAnalysisCode" CHAR(20) NULL,  
"ShiftReferenceCode" CHAR(20) NULL,  
"ShiftSerialNumber" BIGINT NULL,  
"TempShiftTypeID" CHAR(20) NULL,  
"AnalysisBatch" INTEGER NULL,  
"TransferBatch" INTEGER NULL,  
"EmploymentPosition" CHAR(50) NULL,  
"EssentialSkillGradeID" CHAR(4) NULL,
```



```
"vacancytheirref" CHAR(50) NULL,
"vacancyrefcode" CHAR(20) NULL,
"CustomString1" CHAR(40) NULL,
"CustomString2" CHAR(40) NULL,
"CustomString3" CHAR(40) NULL,
"CustomString4" CHAR(40) NULL,
"CustomString5" CHAR(40) NULL,
"CustomDouble1" DOUBLE NULL,
"CustomDouble2" DOUBLE NULL,
"CustomDouble3" DOUBLE NULL,
"CustomDouble4" DOUBLE NULL,
"CustomDouble5" DOUBLE NULL,
PRIMARY KEY("AccountsStatisticsID"),
) ON commit DELETE ROWS;
CREATE INDEX "a" ON "pears"."TEMPAccountsStatistics"("TempTimesheetID");
CREATE INDEX "b" ON "pears"."TEMPAccountsStatistics"("PersonID");
CREATE INDEX "c" ON "pears"."TEMPAccountsStatistics"("PayrollNumber");
CREATE INDEX "d" ON
"pears"."TEMPAccountsStatistics"("TempTimesheetLineID");
CREATE INDEX "e" ON
"pears"."TEMPAccountsStatistics"("NominalCodeSegment1");
CREATE INDEX "f" ON
"pears"."TEMPAccountsStatistics"("NominalCodeSegment2");
CREATE INDEX "g" ON
"pears"."TEMPAccountsStatistics"("NominalCodeSegment3");
CREATE INDEX "h" ON
"pears"."TEMPAccountsStatistics"("NominalCodeSegment4");
CREATE INDEX "i" ON
"pears"."TEMPAccountsStatistics"("NominalCodeSegment5");
CREATE INDEX "j" ON "pears"."TEMPAccountsStatistics"("PlacementID");
CREATE INDEX "l" ON
"pears"."TEMPAccountsStatistics"("TempTimesheetLineID");
CREATE INDEX "k" ON "pears"."TEMPAccountsStatistics"("DocumentID");
BEGIN atomic
--
--
-- Update the Temptimesheet lines with the new Analysis batch
-- if transfer batch > 0 then use this as a filter, ie records in
transfer batch will mirror the transfer batch number
-- if transferbatch = 0, then pick up journal records in the date range
@PeriodFrom to @periodto
--
IF "IsNull"(@TransferBatch,0) > 0 THEN
UPDATE "Temptimesheet" AS "t"
SET "t"."AnalysisBatch" = @AnalysisBatch FROM
"temptimesheet" AS "t" LEFT OUTER JOIN("placement" KEY JOIN
"employment" KEY JOIN "company")
WHERE "isnull"("t"."TransferBatch",0) = @transferbatch AND
```



```
"company"."clientcode" LIKE @AccountFilterLike
AND "isnull"("t"."AnalysisBatch",0) = 0
AND "isnull"("company"."divisionid",'') LIKE @DivFilterLike
AND "isnull"("placement"."departmentid",'') LIKE @DeptFilterLike
ELSE
UPDATE "Temptimesheet" AS "t"
SET "t"."AnalysisBatch" = @AnalysisBatch FROM
"temptimesheet" AS "t" LEFT OUTER JOIN("placement" KEY JOIN
"employment" KEY JOIN "company")
WHERE "company"."clientcode" LIKE @AccountFilterLike AND
"isnull"("t"."AnalysisBatch",0) = 0
AND "date"("t"."whenentered") BETWEEN @FromDate AND @ToDate
AND "isnull"("company"."divisionid",'') LIKE @DivFilterLike
AND "isnull"("placement"."departmentid",'') LIKE @DeptFilterLike
END IF;
--
--
-- Populate temporary table with Data, fixed data first. Should not
needs to change this section
--
--
--
-- timesheet line related details, dont change this section
--
--
INSERT INTO "TEMPAccountsStatistics"( "AccountsStatisticsID",
"TemptimesheetID","TimesheetSerial","TimesheetLineNumber","Personid","Payrol
lNumber","Placementid","Tempdeskid",
"TimesheetAnalysisCode","tempshiftid","paid","PayBandPayRollFlag","IsExpense
s","charged","AnalysisBatch",
"TaxMethod","TempTimesheetLineID","Period","Quantity",
"Unitdescription","accountingdate","transferbatch" )
SELECT "uniquekey"("number"()),
"tl"."TempTimesheetID",
"t"."SerialNumber",
"tl"."LineNumber",
"t"."personid",
"t"."payrollnumber",
"t"."PlacementID",
"t"."TempDeskID",
"t"."AnalysisCode",
"tl"."TempShiftID",
"isnull"("tl"."unitspaid"*"tl"."payrate",0) AS "paid",
"PayrollFlag",
ISNULL("IsExpenses",0),
"isnull"("tl"."UnitsCharged"*"tl"."ChargeRate",0) AS "charged",
"AnalysisBatch","TaxMethod","tl"."TempTimesheetLineID","Period",
```



```
isnull ("unitspaid",0),
"b"."unit","whenentered","t"."transferbatch"
FROM "TempTimesheetLine" AS "tl"
KEY JOIN("TempTimesheet" AS "t","temppayband" AS "b") WHERE
"analysisbatch" = @AnalysisBatch;
--
--

UPDATE "TEMPAccountsStatistics" AS "tas" SET "journalnumber" =
"j"."number","amount" = "j"."amount",
"goodsamount" = "j"."goodsamount","JournalNominalCode" =
"j"."NominalCode",
"journalclass" = "j"."journalclass","priceeach" =
"j"."priceeach","vatamount" = "j"."vatamount","vatcode" = "j"."vatcode",
"description" = "j"."description", "documentid" = "j"."documentid"
FROM
"TEMPAccountsStatistics" AS "tas" JOIN "iqacjournal" AS "j" ON "xref"
= 'T' AND "xrefid" = "tas"."temptimesheetlineid" ;

UPDATE "TEMPAccountsStatistics" AS "tas" SET "documentnumber" =
"ourref","DocumentDate" = d.Documentdate FROM
"TEMPAccountsStatistics" AS "tas" JOIN "iqacddocument" AS "d" ON
"tas"."documentid" = "d"."documentid" AND "d"."ledgerid" = 'SALES';
--
-- Person ID when no timesheet line
--
--
UPDATE "TEMPAccountsStatistics" AS "tas"
SET "vacancyid" = "v"."vacancyid","vacancytheirref" =
"v"."theirref","vacancyrefcode" = "v"."refcode" FROM
"TEMPAccountsStatistics" AS "tas" JOIN "placement" AS "p" ON
"tas"."placementID" = "p"."placementID" KEY JOIN("vacancy" AS
"v","employment" AS "e");
--
--
-- Employment Position, Placement Temp, Currency
--
--
UPDATE "TEMPAccountsStatistics" AS "tas" SET "EmploymentPosition" =
"Employment"."Position",
"currency" = "isnull"("tas"."currency","p"."currency") FROM
"TEMPAccountsStatistics" AS "tas" JOIN "placement" AS "p" ON
"tas"."placementID" = "p"."placementID"
,"placement" AS "p" KEY JOIN "employment";
--
-- Employment Position, Placement Temp, Currency
```



```
--
--
UPDATE "TEMPAccountsStatistics" AS "tas"
  SET "Period" = "t"."Period", "payrollidentifier" =
"t"."payrollidentifier", "TimesheetSupplierCode" = "t"."SupplierCode" FROM
  "TEMPAccountsStatistics" AS "tas" JOIN "TempTimesheet" AS "t" ON
"tas"."TempTimesheetID" = "t"."TempTimesheetID";
--
-- Shift Fields
--
--
UPDATE "TEMPAccountsStatistics" AS "tas"
  SET "ShiftReferenceCode" = "t"."referencecode", "ShiftSerialNumber" =
"p"."shiftserialnumber", "tempshiftdate" = "t"."shiftdate",
  "Shiftanalysiscode" = "t"."AnalysisCode", "EssentialSkillGradeID" =
"t"."EssentialSkillGradeID",
  "TempShifttypeid" =
"t"."TempShifttypeid", "shiftconsultantAnalysisCode" = (SELECT
"s"."analysiscode" FROM "staff" AS "s" WHERE "s"."staffid" = "t"."staffid")
FROM
  "TEMPAccountsStatistics" AS "tas" JOIN "TempShift" AS "t" ON
"tas"."TempShiftID" = "t"."TempShiftID"
  , "TempShift" AS "t" KEY JOIN "TempShiftplan" AS "p";
--
--
-- Person names
--
--
UPDATE "TEMPAccountsStatistics" AS "tas" SET "personname" = "p"."name"
FROM "TEMPAccountsStatistics" AS "tas" JOIN "person" AS "p" ON
"tas"."personid" = "p"."personid";
--
--
-- Company Details for placement related invoices
--
--
UPDATE "TEMPAccountsStatistics" AS "tas"
  SET "CompanyID" = "c"."CompanyID",
  "CompanyName" = "c"."Name",
  "ClientAccountCode" = "c"."clientcode" FROM
  "TEMPAccountsStatistics" AS "tas" JOIN "placement" AS "p" ON
"tas"."placementID" = "p"."placementID"
  KEY JOIN "vacancy" KEY JOIN "employment" AS "e" KEY JOIN "company" AS
"c";
--
--
-- Placement Nominal Codes
```



```
--
--
UPDATE "TEMPAccountsStatistics" AS "tas" SET "NominalCodeSegment1" =
(SELECT FIRST "nominalcode"
FROM(SELECT top 1 "nominalcode" FROM "placementanalysis" AS "pl1"
,(SELECT "count"() AS "plc" FROM "placementanalysis" AS "pl"
WHERE "pl"."placementid" = "tas"."placementid") AS "t2"
WHERE "pl1"."placementid" = "tas"."placementid" AND "plc" >
ORDER BY "whenauthorised" ASC) AS "t1" ORDER BY "nominalcode" DESC) WHERE
"placementid" IS NOT NULL;
UPDATE "TEMPAccountsStatistics" AS "tas"
SET "NominalCodeSegment2"
= (SELECT FIRST "nominalcode"
FROM(SELECT top 2 "nominalcode" FROM "placementanalysis" AS "pl1"
,(SELECT "count"() AS "plc" FROM "placementanalysis" AS "pl"
WHERE "pl"."placementid" = "tas"."placementid") AS "t2"
WHERE "pl1"."placementid" = "tas"."placementid" AND "plc" > 1
ORDER BY "whenauthorised" ASC) AS "t1" ORDER BY "nominalcode" DESC) WHERE
"placementid" IS NOT NULL AND "NominalCodeSegment1" IS NOT NULL;
UPDATE "TEMPAccountsStatistics" AS "tas" SET "NominalCodeSegment3" =
(SELECT FIRST "nominalcode"
FROM(SELECT top 3 "nominalcode" FROM "placementanalysis" AS "pl1"
,(SELECT "count"() AS "plc" FROM "placementanalysis" AS "pl"
WHERE "pl"."placementid" = "tas"."placementid") AS "t2"
WHERE "pl1"."placementid" = "tas"."placementid" AND "plc" > 2
ORDER BY "whenauthorised" ASC) AS "t1" ORDER BY "nominalcode" DESC) WHERE
"placementid" IS NOT NULL AND "NominalCodeSegment2" IS NOT NULL;
UPDATE "TEMPAccountsStatistics" AS "tas" SET "NominalCodeSegment4" =
(SELECT FIRST "nominalcode"
FROM(SELECT top 4 "nominalcode" FROM "placementanalysis" AS "pl1"
,(SELECT "count"() AS "plc" FROM "placementanalysis" AS "pl"
WHERE "pl"."placementid" = "tas"."placementid") AS "t2"
WHERE "pl1"."placementid" = "tas"."placementid" AND "plc" > 3
ORDER BY "whenauthorised" ASC) AS "t1" ORDER BY "nominalcode" DESC) WHERE
"placementid" IS NOT NULL AND "NominalCodeSegment3" IS NOT NULL;
UPDATE "TEMPAccountsStatistics" AS "tas" SET "NominalCodeSegment5" =
(SELECT FIRST "nominalcode"
FROM(SELECT top 5 "nominalcode" FROM "placementanalysis" AS "pl1"
,(SELECT "count"() AS "plc" FROM "placementanalysis" AS "pl"
WHERE "pl"."placementid" = "tas"."placementid") AS "t2"
WHERE "pl1"."placementid" = "tas"."placementid" AND "plc" > 4
ORDER BY "whenauthorised" ASC) AS "t1" ORDER BY "nominalcode" DESC) WHERE
"placementid" IS NOT NULL AND "NominalCodeSegment4" IS NOT NULL;
--
--
-- StaffID
--
--
```



```
UPDATE "TEMPAccountsStatistics" AS "tas" SET "TimesheetStaffID" =
(SELECT "staffid" FROM "temptimesheet" AS "t" WHERE "t"."temptimesheetid" =
"tas"."temptimesheetid"),
    "PlacementStaffID" = (SELECT "staffid" FROM "placement" AS "p" WHERE
"p"."placementid" = "tas"."placementid"),
    "vacancyStaffID" = (SELECT "staffid" FROM "vacancy" AS "v" WHERE
"v"."vacancyid" = "tas"."vacancyid");
--
--
-- Analysis Codes
--
--
UPDATE "TEMPAccountsStatistics" AS "tas"
    SET "DivisionAnalysisCode" = "isnull"((SELECT "Analysis" FROM
"Division" KEY JOIN "TempDesk" WHERE "TempDesk"."TempDeskID" =
"tas"."TempDeskID"),
    (SELECT "Analysis" FROM "Division" KEY JOIN "Company" WHERE
"tas"."CompanyID" = "Company"."CompanyID")),
    "DepartmentAnalysisCode" = (SELECT "Analysis" FROM "Department" KEY
JOIN "Placement" WHERE "Placement"."PlacementID" = "tas"."PlacementID"),
    "TempdeskAnalysiscode" = "isnull"((SELECT "Defanalysiscode" FROM
"TempDesk" WHERE "TempDesk"."TempDeskID" = "tas"."TempDeskID"),''),
    "VacancyConsultantAnalysisCode" = (SELECT "s"."analysiscode" FROM
"staff" AS "s" WHERE "tas"."vacancystaffid" = "s"."staffid"),
    "PlacementConsultantAnalysisCode" = (SELECT "s"."analysiscode" FROM
"staff" AS "s" WHERE "tas"."placementstaffid" = "s"."staffid"),
    "TimeSheetConsultantAnalysisCode" = (SELECT "s"."analysiscode" FROM
"staff" AS "s" WHERE "tas"."timesheetstaffid" = "s"."staffid"),
    "ConsultantAnalysisCode" = (SELECT "s"."analysiscode" FROM "person"
KEY JOIN "staff" AS "s" WHERE "tas"."personid" = "person"."personid");
    UPDATE "TEMPAccountsStatistics" AS "tas" SET "personname" = "p"."name"
FROM "TEMPAccountsStatistics" AS "tas" JOIN "person" AS "p" ON
"tas"."personid" = "p"."personid";
--
--
-- Now the extras which WILL different per site
--
--un
--
-- Pensions, Holiday, Levy
--
UPDATE "TEMPAccountsStatistics" AS "tas"
    SET "HolPay" = IF "IncludeInHolidayPay" = '1' THEN "paid" ELSE 0 endif
FROM
    "TEMPAccountsStatistics" AS "tas" JOIN "temptimesheetline" AS "l"
ON("l"."temptimesheetlineid" =
"tas"."temptimesheetlineid"), "temptimesheetline" AS "l" KEY JOIN
"temppayband" WHERE "tas"."temptimesheetlineid"
```



```
= "l"."temptimesheetlineid";
UPDATE "TEMPAccountsStatistics" AS "tas"
SET "AWRHolRate"
= IF "isnull"("AWRWasInvolved",0) = 0 THEN
    "isnull"((SELECT FIRST("ExtraHols"/(52-"ExtraHols")) FROM
"AWRJobMaster" AS "a" WHERE "a"."placementid" = "tas"."placementid"),0) ELSE
0 endif FROM
    "TEMPAccountsStatistics" AS "tas" JOIN "temptimesheet" AS "t" ON
"tas"."temptimesheetid" = "t"."temptimesheetid";
-- update "TEMPAccountsStatistics" as "tas" set "HolLiab" =
"PayRate"*"UnitsPaid" from
-- "temptimesheetline" key join "temppayband" as "b" where
"b"."description" like '%HP';
-- update "TEMPAccountsStatistics" as "tas" set "Levy" =
"isnull"((select "sum"("amount") from "accord"."IQXJournalHistory" where
"trim"("cosegmenta") = "timesheetserial" and("wtcode" = '603950')),0);
-- update "TEMPAccountsStatistics" as "tas" set "HolLiabNI" =
"iqmoneyround"("HolLiab"*.1);
-- update "TEMPAccountsStatistics" as "tas" set "ActualNI" =
"isnull"((select "sum"("amount") from "accord"."IQXJournalHistory" where
"trim"("cosegmenta") = "timesheetserial" and(("wtcode" = '603000')
or("wtcode" = '603500'))),0);
-- update "TEMPAccountsStatistics" as "tas" set "Pension" =
"isnull"((select "sum"("amount") from "accord"."IQXJournalHistory" where
"trim"("cosegmenta") = "timesheetserial" and("wtcode" = '540620')),0);
UPDATE "TEMPAccountsStatistics" AS "tas" SET "Statutory" = 0;
UPDATE "TEMPAccountsStatistics" AS "tas" SET "Charged" =
"Charged"+"Pension"
WHERE EXISTS(SELECT * FROM "placement" AS "p" KEY JOIN "employment"
KEY JOIN "companyaccount" WHERE "p"."placementid" = "tas"."placementid" AND
"ERNIoninvoice" = 1)
AND EXISTS(SELECT * FROM "temptimesheetline" AS "l" WHERE
"l"."temptimesheetlineid" = "tas"."temptimesheetlineid" AND "l"."linenumber"
= 1);
--
-- questions etc, WILL different per site
--
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomDouble1" =
(SELECT("unitscharged"*"chargerate")-(("unitspaid"*"payrate") FROM
"temptimesheetline" AS "t" WHERE "tas"."temptimesheetlineid" =
"t"."temptimesheetlineid");
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomDouble2" =
(("paid"/"charged")*100) WHERE "charged" <> 0;
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomDouble3" = "paid";
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomDouble4" =
"charged";
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomDouble5" = 0;
UPDATE "TEMPAccountsStatistics" SET "type" = '';
```



```
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomString1" = (SELECT
"department"."name" FROM "placement" KEY JOIN "department" WHERE
"placement"."placementid" = "tas"."placementid");
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomString2" = (SELECT
"staff"."userid" FROM "staff" WHERE "staff"."staffid" =
"tas"."vacancystaffid");
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomString3" = (SELECT
"keyname" FROM "person" WHERE "person"."personid" = "tas"."personid");
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomString4" = (SELECT
"surname" FROM "person" WHERE "person"."personid" = "tas"."personid");
UPDATE "TEMPAccountsStatistics" AS "tas" SET "CustomString5" = (SELECT
"forenames" FROM "person" WHERE "person"."personid" = "tas"."personid");
--
--
-- Finally Insert the temp table into the Main Accounts Stats table - Do
not modify
--
--
--
--
-- delete from "TEMPAccountsStatistics";
--
--
-- Insert the Analysis Batch into the batch transfer table- Do not
modify
INSERT INTO "IQXAccountsStatistics"
(
"IQXAccountsStatisticsID", "AccountingDate", "Amount", "Description", "Type",
"NominalCodeSegment1", "NominalCodeSegment2", "NominalCodeSegment3", "NominalCo
deSegment4", "NominalCodeSegment5", "PayrollCompany",
"DivisionAnalysisCode", "DepartmentAnalysisCode", "TimesheetAnalysisCode", "Tim
esheetConsultantAnalysisCode", "VacancyConsultantAnalysisCode", "PlacementCons
ultantAnalysisCode", "ConsultantAnalysisCode",
"ClientAccountCode", "CompanyID", "PersonID",
"TempTimesheetID", "TempTimesheetLineID", "TempDeskID", "VacancyStaffID", "Place
mentStaffid", "TimeSheetStaffid", "PayrollNumber", "PersonName", "CompanyName", "
TimesheetSerial",
"TimesheetLineNumber", "DocumentNumber", "DocumentID", "DocumentDate", "JournalL
ineNumber", "SalesOrCost", "Currency", "PlacementID", "VacancyID", "TempShiftID",
"tempshiftdate", "DocumentLineType",
"JournalClass", "Quantity", "Unitdescription", "TaxMethod", "PaID", "HolPay", "Hol
Liab", "AWRHolRate", "Levy",
"HolLiabNI", "ActualNI", "Pension", "Statutory", "Charged", "TempdeskAnalysiscode
", "GoodsAmount", "priceEach", "VatAmount", "VatCode", "JournalNominalCode",
"PayrollIdentifier", "Period", "TimesheetSupplierCode", "ShiftAnalysisCode", "Sh
iftReferenceCode", "ShiftSerialNumber", "TempShiftTypeID",
"CustomString1", "CustomString2", "CustomString3", "CustomString4", "CustomStrin
g5", "CustomDouble1", "CustomDouble2", "CustomDouble3",
```



```
"CustomDouble4","CustomDouble5","AnalysisBatch","EmploymentPosition","EssentialSkillGradeID","PayBandPayrollFlag","IsExpenses","VacancyTheirRef","VacancyRefcode","ShiftConsultantAnalysisCode",
    transferbatch)
SELECT
"AccountsStatisticsID","AccountingDate","Amount","Description","Type",
"NominalCodeSegment1","NominalCodeSegment2","NominalCodeSegment3","NominalCodeSegment4","NominalCodeSegment5","PayrollCompany",
"DivisionAnalysisCode","DepartmentAnalysisCode","TimesheetAnalysisCode","TimesheetConsultantAnalysisCode","VacancyConsultantAnalysisCode","PlacementConsultantAnalysisCode","ConsultantAnalysisCode",
    "ClientAccountCode","CompanyID","PersonID",
"TempTimesheetID","TempTimesheetLineID","TempDeskID","VacancyStaffID","PlacementStaffID","TimeSheetStaffid","PayrollNumber","PersonName","CompanyName","TimesheetSerial",
"TimesheetLineNumber","DocumentNumber","DocumentID","DocumentDate","JournalLineNumber","SalesOrCost","Currency","PlacementID","VacancyID","TempShiftID",
"tempshiftdate","DocumentLineType",
"JournalClass","Quantity","Unitdescription","TaxMethod","PaID","HolPay","HolLiab","AWRHolRate","Levy",
"HolLiabNI","ActualNI","Pension","Statutory","Charged","TempdeskAnalysiscode",
"GoodsAmount","priceEach","VatAmount","VatCode","JournalNominalCode",
"PayrollIdentifier","Period","TimesheetSupplierCode","ShiftAnalysisCode","ShiftReferenceCode","ShiftSerialNumber","TempShiftTypeID",
"CustomString1","CustomString2","CustomString3","CustomString4","CustomString5","CustomDouble1","CustomDouble2","CustomDouble3",
"CustomDouble4","CustomDouble5","AnalysisBatch","EmploymentPosition","EssentialSkillGradeID","PayBandPayrollFlag","IsExpenses","VacancyTheirRef","VacancyRefcode","ShiftConsultantAnalysisCode",
    transferbatch
FROM "TEMPAccountsStatistics";
--
INSERT INTO "BatchTransfers"(
"BatchType","BatchNumber","TransferTime","Note" )
VALUES( 'IQXAnalysis',@AnalysisBatch,CURRENT_TIMESTAMP,string('Accounts Statistics Export by ', (SELECT userid FROM staff WHERE staffid =
userstaffid ))) ;
-- drop the temporary table
DROP TABLE "pears"."TEMPAccountsStatistics"
-- return the transfer batch
-- return(@AnalysisBatch)
END // END OF atomic statement
END
}
```



From:

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

Permanent link:

https://iqxusers.co.uk/iqxhelp/doku.php?id=database:procedures:pears_iqxpopulateaccountstatisticsiqac

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

