**1. Complete Insert Statements with Testing Selects**

sql

Copy

-- 1. Masterlist Insert

INSERT INTO #ArchRbMasterlist (

MmsStockCode, ItemRefNumber, DetailedDesc, Keyword,

CorePartNum, RebuiltStockNum, CoreCharge, JobNumber,

EstimatedCost, SopNumber, DateArchived, ArchivedBy,

LabourCost, LabourOverheadAmount, LabourTotal,

MaterialsTotal, TotalRebuiltPartCost, TaxTotal, TotalCostTax

)

SELECT

RM.MmsStockCode,

RM.ItemRefNumber,

RM.DetailedDesc,

RM.Keyword,

RM.CorePartNum,

RM.RebuiltStockNum,

RM.CoreCharge,

RM.JobNumber,

RM.EstimatedCost,

RM.SopNumber,

GETDATE(),

@ViUsername,

@ViLabourCost,

@ViLabourOverheadAmount,

@ViLabourTotal,

@ViMaterialsTotal,

@ViTotalRebuiltPartCost,

@ViTaxTotal,

@ViTotalCostTax

FROM SBCES.RbMasterlist RM

WHERE RM.RebuiltStockNum = @ViRebuiltStockNum;

SET @ArchId = SCOPE\_IDENTITY();

SELECT \* FROM #ArchRbMasterlist;

-- 2. Employee Labour

INSERT INTO #ArchRbEmployeeLabour (

LabourDefn, DateEntered, LinkNumber, TypeId,

CostCentre, Task, LabourType, [Usage],

HrsReqd, AdjHrs, TimeAddition, ArchRbMasterlistId

)

SELECT

EL.LabourDefn,

EL.DateEntered,

EL.LinkNumber,

EL.TypeId,

EL.CostCentre,

EL.Task,

EL.LabourType,

TRY\_CAST(EL.[Usage] AS DECIMAL(5,2)),

EL.HrsReqd,

EL.AdjHrs,

EL.TimeAddition,

@ArchId

FROM SBCES.EmployeeLabour EL

WHERE EL.LinkNumber = @ViLinkNumber

AND EL.TypeId = @ViTypeId;

SELECT \* FROM #ArchRbEmployeeLabour;

-- 3. SC Parts Used (Rebuilt)

INSERT INTO #ArchRbScPartsUsed (

MmsStockCode, DateEntered, RebPartCost, UserEntered,

MmsCost, OemCost, MmsRebuiltCode, CostCentre,

QtyReqd, PercentUsage, LinkCode, RebuiltPart,

LinkType, CoreCost, ArchRbMasterlistId, TotalCost,

ScKeyword, ScDescription

)

SELECT

UPPER(SU.MmsStockCode),

SU.DateEntered,

ISNULL(A.RbUnitCost, 0),

UPPER(SU.UserEntered),

ISNULL(SU.MmsCost, 0),

ISNULL(SU.OemCost, 0),

UPPER(SU.MmsRebuiltCode),

UPPER(SU.CostCentre),

TRY\_CAST(SU.QtyReqd AS DECIMAL(18,2)),

TRY\_CAST(REPLACE(SU.PercentUsage, ',', '.') AS DECIMAL(5,2)),

@ViLinkNumber,

'Y',

'RB',

SU.CoreCost,

@ArchId,

ISNULL(A.RbCost, 0),

UPPER(RM.Keyword),

UPPER(RM.DetailedDesc)

FROM SBCES.ScPartsUsed SU

LEFT JOIN SBCES.RbMasterlist RM

ON RM.RebuiltStockNum = SU.MmsRebuiltCode

LEFT JOIN (

SELECT

RbReference,

ROUND(SUM(TotalCost), 2) AS RbCost,

ROUND(SUM(TotalUnitCost), 2) AS RbUnitCost

FROM SBCES.TempMaterialsList

GROUP BY RbReference

) A ON A.RbReference = SU.MmsRebuiltCode

WHERE SU.LinkCode = @ViLinkNumber

AND SU.RebuiltPart = 'Y'

AND SU.LinkType = @ViTypeId;

SELECT \* FROM #ArchRbScPartsUsed WHERE RebuiltPart = 'Y';

-- 4. SC Parts Used (Non-Rebuilt)

INSERT INTO #ArchRbScPartsUsed (

MmsStockCode, DateEntered, RebPartCost, UserEntered,

MmsCost, OemCost, MmsRebuiltCode, CostCentre,

QtyReqd, PercentUsage, LinkCode, RebuiltPart,

LinkType, CoreCost, ArchRbMasterlistId, TotalCost,

ScKeyword, ScDescription

)

SELECT

A.MmsStockCode,

A.DateEntered,

A.RebPartCost,

A.UserEntered,

A.MmsCost,

A.OemCost,

A.MmsRebuiltCode,

A.CostCentre,

TRY\_CAST(A.QtyReqd AS DECIMAL(18,2)),

TRY\_CAST(REPLACE(A.PercentUsage, ',', '.') AS DECIMAL(5,2)),

@ViLinkNumber,

'N',

@ViTypeId,

SBCES.FinalCost(

TRY\_CAST(REPLACE(A.PercentUsage, ',', '.') AS DECIMAL(5,2)),

TRY\_CAST(A.QtyReqd AS DECIMAL(18,2)),

A.CoreCost,

0

),

@ArchId,

A.MmsCost \* TRY\_CAST(A.QtyReqd AS DECIMAL(18,2))

\* (TRY\_CAST(REPLACE(ISNULL(A.PercentUsage, '0'), ',', '.') AS DECIMAL(5,2))/100),

ISNULL(B.PartType, ''),

ISNULL(B.DetailedDesc, '')

FROM SBCES.ScPartsUsed A

LEFT JOIN SBCES.StockCodedParts B

ON B.MmsStockCode = A.MmsStockCode

WHERE A.LinkCode = @ViLinkNumber

AND A.LinkType = @ViTypeId

AND A.RebuiltPart = 'N';

SELECT \* FROM #ArchRbScPartsUsed WHERE RebuiltPart = 'N';

-- 5. NSC Parts Used

INSERT INTO #ArchRbNscPartsUsed (

OrigSuppNum, OrigSupplierName, CostCentre,

QtyReqd, PercentUsage, DateEntered, EnteredBy,

LinkCode, Cost, LinkType, CoreCost,

ArchRbMasterlistId, TotalCost

)

SELECT

NP.OrigSuppNum,

NP.OrigSupplierName,

NP.CostCentre,

TRY\_CAST(NP.QtyReqd AS DECIMAL(18,2)),

TRY\_CAST(REPLACE(NP.PercentUsage, ',', '.') AS DECIMAL(5,2)),

NP.DateEntered,

NP.EnteredBy,

NP.LinkCode,

NP.Cost,

NP.LinkType,

SBCES.FinalCost(

TRY\_CAST(REPLACE(NP.PercentUsage, ',', '.') AS DECIMAL(5,2)),

TRY\_CAST(NP.QtyReqd AS DECIMAL(18,2)),

NP.CoreCost,

0

),

@ArchId,

NP.Cost \* TRY\_CAST(NP.QtyReqd AS DECIMAL(18,2))

\* (TRY\_CAST(REPLACE(ISNULL(NP.PercentUsage, '0'), ',', '.') AS DECIMAL(5,2))/100)

FROM SBCES.NscPartsUsed NP

WHERE NP.LinkCode = @ViLinkNumber

AND NP.LinkType = @ViTypeId;

SELECT \* FROM #ArchRbNscPartsUsed;

-- 6. Labour CC Total

INSERT INTO #ArchRbLabourCcTotal (

CostCentre, TotalHours, ArchRbMasterlistId

)

SELECT

CC,

SUM(TRY\_CAST(ExtendedTime AS DECIMAL(18,2))),

@ArchId

FROM SBCES.TempLabourList

GROUP BY CC;

SELECT \* FROM #ArchRbLabourCcTotal;

-- 7. Materials CC Total

INSERT INTO #ArchRbMaterialsCcTotal (

CostCentre, TotalCost, ArchRbMasterlistId

)

SELECT

CostCentre,

SUM(TRY\_CAST(TotalCost AS DECIMAL(18,2))),

@ArchId

FROM SBCES.TempMaterialsList

GROUP BY CostCentre;

SELECT \* FROM #ArchRbMaterialsCcTotal;

-- 8. List Buses

INSERT INTO #ArchRbListBuses (

BusDescription, ArchRbMasterlistId

)

SELECT

LB.Description,

@ArchId

FROM SBCES.RbListOfBuses RB

INNER JOIN SBCES.ListOfBuses LB

ON RB.ListId = LB.ListId

WHERE RB.RebuiltStockNum = @ViLinkNumber;

SELECT \* FROM #ArchRbListBuses;

**2. How to Test**

1. **Prepare Test Data**

sql

Copy

-- In production environment, replace with actual test data

INSERT INTO SBCES.RbMasterlist (RebuiltStockNum, MmsStockCode, ...)

VALUES ('TEST123', 'MMS001', ...);

INSERT INTO SBCES.EmployeeLabour (LinkNumber, TypeId, HrsReqd, ...)

VALUES ('LINK456', 'RB', '08:30', ...);

1. **Run Tests**  
   Execute these in order:

sql

Copy

-- A. Run temp table setup script first

-- B. Execute main test script

-- C. After each INSERT, verify results:

1. **Validation Checks**
   * **Row Counts**:

sql

Copy

SELECT 'Master' AS Table, COUNT(\*) FROM #ArchRbMasterlist

UNION ALL

SELECT 'Employee Labour', COUNT(\*) FROM #ArchRbEmployeeLabour;

* + **Cost Calculations**:

sql

Copy

SELECT

LabourTotal AS CalculatedLabour,

LabourCost + LabourOverheadAmount AS ExpectedLabourTotal

FROM #ArchRbMasterlist;

* + **Foreign Key Consistency**:

sql

Copy

SELECT COUNT(\*) AS OrphanRecords

FROM #ArchRbScPartsUsed

WHERE ArchRbMasterlistId NOT IN (

SELECT ArchRbMasterlistId FROM #ArchRbMasterlist

);

1. **Edge Cases**

sql

Copy

-- Test with invalid hours format

UPDATE SBCES.EmployeeLabour

SET HrsReqd = 'AB:CD'

WHERE LinkNumber = 'LINK456';

-- Rerun test script and check error handling

-- Test with missing tax rate

DELETE FROM SBCES.CesSettings

WHERE Parameter = 'TAX\_RATE';

-- Rerun and verify @TaxTotal becomes 0