**Query 1: Search NSC RB Part List**

sql

Copy

-- Search for parts in NSC\_PARTS\_USED, NON\_STOCK\_CODED\_PARTS, and RB\_MASTERLIST based on cost comparison type

IF (vi\_cost\_comparetype = 0) THEN

SELECT

UPPER(RBMASTERLIST.rebuiltstocknum) AS "RebuiltNumber",

UPPER(RBMASTERLIST.mmsstockcode) AS "MMSStockCode",

UPPER(RBMASTERLIST.keyword) AS "Keyword",

UPPER(RBMASTERLIST.detaileddesc) AS "Description",

RBMASTERLIST.rowid

FROM

NSCPARTSUSED,

NONSTOCKCODEDPARTS,

RBMASTERLIST

WHERE

NVL(NSCPARTSUSED.origsuppnum, ' ') LIKE TRIM(vi\_orig\_supplier\_num)

AND UPPER(NVL(NONSTOCKCODEDPARTS.detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND NSCPARTSUSED.origsuppnum = NONSTOCKCODEDPARTS.origsuppliernum

AND NSCPARTSUSED.linkcode = RBMASTERLIST.rebuiltstocknum

AND NONSTOCKCODEDPARTS.oempartcost <= vi\_oem\_part\_cost

AND NSCPARTSUSED.linktype = 'RB';

ELSIF (vi\_cost\_comparetype = 1) THEN

SELECT

UPPER(RBMASTERLIST.rebuiltstocknum) AS "RebuiltNumber",

UPPER(RBMASTERLIST.mmsstockcode) AS "MMSStockCode",

UPPER(RBMASTERLIST.keyword) AS "Keyword",

UPPER(RBMASTERLIST.detaileddesc) AS "Description",

RBMASTERLIST.rowid

FROM

NSCPARTSUSED,

NONSTOCKCODEDPARTS,

RBMASTERLIST

WHERE

NVL(NSCPARTSUSED.origsuppnum, ' ') LIKE TRIM(vi\_orig\_supplier\_num)

AND UPPER(NVL(NONSTOCKCODEDPARTS.detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND NSCPARTSUSED.origsuppnum = NONSTOCKCODEDPARTS.origsuppliernum

AND NSCPARTSUSED.linkcode = RBMASTERLIST.rebuiltstocknum

AND NONSTOCKCODEDPARTS.oempartcost = vi\_oem\_part\_cost

AND NSCPARTSUSED.linktype = 'RB';

ELSIF (vi\_cost\_comparetype = 2) THEN

SELECT

UPPER(RBMASTERLIST.rebuiltstocknum) AS "RebuiltNumber",

UPPER(RBMASTERLIST.mmsstockcode) AS "MMSStockCode",

UPPER(RBMASTERLIST.keyword) AS "Keyword",

UPPER(RBMASTERLIST.detaileddesc) AS "Description",

RBMASTERLIST.rowid

FROM

NSCPARTSUSED,

NONSTOCKCODEDPARTS,

RBMASTERLIST

WHERE

NVL(NSCPARTSUSED.origsuppnum, ' ') LIKE TRIM(vi\_orig\_supplier\_num)

AND UPPER(NVL(NONSTOCKCODEDPARTS.detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND NSCPARTSUSED.origsuppnum = NONSTOCKCODEDPARTS.origsuppliernum

AND NSCPARTSUSED.linkcode = RBMASTERLIST.rebuiltstocknum

AND NONSTOCKCODEDPARTS.oempartcost >= vi\_oem\_part\_cost

AND NSCPARTSUSED.linktype = 'RB';

END IF;

**Query 2: Search NSC Part**

sql

Copy

-- Search for parts in NONSTOCKCODEDPARTS based on cost comparison type

IF (vi\_cost\_comparetype = 0) THEN

SELECT

UPPER(origsuppliernum) AS "SupplierNumber",

UPPER(NVL(origsuppliername, ' ')) AS "SupplierName",

UPPER(NVL(detaileddesc, ' ')) AS "Description",

NVL(oempartcost, 0) AS "Price",

NVL(keyword, ' ') AS "Keyword",

lastmodifieddate AS "LastModifiedDate",

ID AS "ID",

ROWID

FROM

NONSTOCKCODEDPARTS

WHERE

NVL(origsuppliernum, ' ') LIKE vi\_orig\_supplier\_num || '%'

AND UPPER(NVL(detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND UPPER(NVL(keyword, ' ')) LIKE '%' || TRIM(UPPER(vi\_keyword)) || '%'

AND oempartcost <= vi\_oem\_part\_cost

ORDER BY origsuppliernum ASC;

ELSIF (vi\_cost\_comparetype = 1) THEN

SELECT

UPPER(origsuppliernum) AS "SupplierNumber",

UPPER(NVL(origsuppliername, ' ')) AS "SupplierName",

UPPER(NVL(detaileddesc, ' ')) AS "Description",

NVL(oempartcost, 0) AS "Price",

NVL(keyword, ' ') AS "Keyword",

lastmodifieddate AS "LastModifiedDate",

ID AS "ID",

ROWID

FROM

NONSTOCKCODEDPARTS

WHERE

NVL(origsuppliernum, ' ') LIKE vi\_orig\_supplier\_num || '%'

AND UPPER(NVL(detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND UPPER(NVL(keyword, ' ')) LIKE '%' || TRIM(UPPER(vi\_keyword)) || '%'

AND oempartcost = vi\_oem\_part\_cost

ORDER BY origsuppliernum ASC;

ELSIF (vi\_cost\_comparetype = 2) THEN

SELECT

UPPER(origsuppliernum) AS "SupplierNumber",

UPPER(NVL(origsuppliername, ' ')) AS "SupplierName",

UPPER(NVL(detaileddesc, ' ')) AS "Description",

NVL(oempartcost, 0) AS "Price",

NVL(keyword, ' ') AS "Keyword",

lastmodifieddate AS "LastModifiedDate",

ID AS "ID",

ROWID

FROM

NONSTOCKCODEDPARTS

WHERE

NVL(origsuppliernum, ' ') LIKE vi\_orig\_supplier\_num || '%'

AND UPPER(NVL(detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND UPPER(NVL(keyword, ' ')) LIKE '%' || TRIM(UPPER(vi\_keyword)) || '%'

AND oempartcost >= vi\_oem\_part\_cost

ORDER BY origsuppliernum ASC;

END IF;

**Query 3: Search SC Part**

sql

Copy

-- Search for parts in STOCKCODEDPARTS based on cost comparison type

IF (vi\_cost\_comparetype = 0) THEN

SELECT

UPPER(mmsstockcode) AS "StockCode",

UPPER(NVL(detaileddesc, ' ')) AS "Description",

NVL(oempartcost, 0) AS "OEMPartCost",

NVL(mmsnewcost, 0) AS "Price",

UPPER(NVL(parttype, ' ')) AS "Keyword",

lastmodifieddate AS "LastModifiedDate",

ROWID

FROM

STOCKCODEDPARTS

WHERE

NVL(mmsstockcode, ' ') LIKE '%' || TRIM(vi\_orig\_supplier\_num) || '%'

AND UPPER(NVL(detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND UPPER(NVL(parttype, ' ')) LIKE '%' || TRIM(UPPER(vi\_keyword)) || '%'

AND mmsnewcost <= vi\_oem\_part\_cost

ORDER BY mmsstockcode;

ELSIF (vi\_cost\_comparetype = 1) THEN

SELECT

UPPER(mmsstockcode) AS "StockCode",

UPPER(NVL(detaileddesc, ' ')) AS "Description",

NVL(oempartcost, 0) AS "OEMPartCost",

NVL(mmsnewcost, 0) AS "Price",

UPPER(NVL(parttype, ' ')) AS "Keyword",

lastmodifieddate AS "LastModifiedDate",

ROWID

FROM

STOCKCODEDPARTS

WHERE

NVL(mmsstockcode, ' ') LIKE '%' || TRIM(vi\_orig\_supplier\_num) || '%'

AND UPPER(NVL(detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND UPPER(NVL(parttype, ' ')) LIKE '%' || TRIM(UPPER(vi\_keyword)) || '%'

AND mmsnewcost = vi\_oem\_part\_cost

ORDER BY mmsstockcode;

ELSIF (vi\_cost\_comparetype = 2) THEN

SELECT

UPPER(mmsstockcode) AS "StockCode",

UPPER(NVL(detaileddesc, ' ')) AS "Description",

NVL(oempartcost, 0) AS "OEMPartCost",

NVL(mmsnewcost, 0) AS "Price",

UPPER(NVL(parttype, ' ')) AS "Keyword",

lastmodifieddate AS "LastModifiedDate",

ROWID

FROM

STOCKCODEDPARTS

WHERE

NVL(mmsstockcode, ' ') LIKE '%' || TRIM(vi\_orig\_supplier\_num) || '%'

AND UPPER(NVL(detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND UPPER(NVL(parttype, ' ')) LIKE '%' || TRIM(UPPER(vi\_keyword)) || '%'

AND (mmsnewcost >= vi\_oem\_part\_cost OR mmsnewcost IS NULL)

ORDER BY mmsstockcode;

END IF;

**Query 4: Search RB Part**

sql

Copy

-- Search for parts in RBMASTERLIST based on cost comparison type

IF (vi\_cost\_comparetype = 0) THEN

SELECT

UPPER(rebuiltstocknum) AS "RebuiltNumber",

UPPER(mmsstockcode) AS "MMSStockCode",

UPPER(keyword) AS "Keyword",

UPPER(detaileddesc) AS "Description",

ROWID

FROM

RBMASTERLIST

WHERE

UPPER(NVL(rebuiltstocknum, ' ')) LIKE '%' || TRIM(vi\_orig\_supplier\_num) || '%'

AND UPPER(NVL(detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND UPPER(NVL(keyword, ' ')) LIKE '%' || TRIM(UPPER(vi\_keyword)) || '%'

AND estimatedcost <= vi\_oem\_part\_cost;

ELSIF (vi\_cost\_comparetype = 1) THEN

SELECT

UPPER(rebuiltstocknum) AS "RebuiltNumber",

UPPER(mmsstockcode) AS "MMSStockCode",

UPPER(keyword) AS "Keyword",

UPPER(detaileddesc) AS "Description",

ROWID

FROM

RBMASTERLIST

WHERE

UPPER(NVL(rebuiltstocknum, ' ')) LIKE '%' || TRIM(vi\_orig\_supplier\_num) || '%'

AND UPPER(NVL(detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND UPPER(NVL(keyword, ' ')) LIKE '%' || TRIM(UPPER(vi\_keyword)) || '%'

AND estimatedcost = vi\_oem\_part\_cost;

ELSIF (vi\_cost\_comparetype = 2) THEN

SELECT

UPPER(rebuiltstocknum) AS "RebuiltNumber",

UPPER(mmsstockcode) AS "MMSStockCode",

UPPER(keyword) AS "Keyword",

UPPER(detaileddesc) AS "Description",

ROWID

FROM

RBMASTERLIST

WHERE

UPPER(NVL(rebuiltstocknum, ' ')) LIKE '%' || TRIM(vi\_orig\_supplier\_num) || '%'

AND UPPER(NVL(detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND UPPER(NVL(keyword, ' ')) LIKE '%' || TRIM(UPPER(vi\_keyword)) || '%'

AND estimatedcost >= vi\_oem\_part\_cost;

END IF;

**Query 5: Retrieve Vehicle Types**

sql

Copy

-- Retrieve all vehicle types from TYPEOFVEHICLE

SELECT

UPPER(make) AS "Make",

UPPER(model) AS "Model",

UPPER(year) AS "Year",

ROWID

FROM

TYPEOFVEHICLE;

**Query 6: Retrieve Vehicle Estimates**

sql

Copy

-- Retrieve all vehicle estimates from VEHICLELIST

SELECT

UPPER(vehiclelistid) AS "VehicleListId",

UPPER(vehseriescode) AS "VehSeriesCode",

UPPER(numofvehicles) AS "NumOfVehicles",

UPPER(projdesc) AS "ProjDesc",

UPPER(dateentered) AS "DateEntered",

UPPER(enteredby) AS "EnteredBy",

UPPER(modifiedlastby) AS "ModifiedLastBy",

UPPER(modifiedlastdate) AS "ModifiedLastDate",

UPPER(make) AS "Make",

UPPER(model) AS "Model",

UPPER(year) AS "Year",

UPPER(engine) AS "Engine",

UPPER(transmission) AS "Transmission",

UPPER(differential) AS "Differential",

UPPER(sopnumber) AS "SopNumber",

ROWID

FROM

VEHICLELIST;

**Query 7: Retrieve Engines**

sql

Copy

-- Retrieve all engines from ENGINES

SELECT

UPPER(name) AS "Name",

ROWID

FROM

ENGINES;

**Query 8: Retrieve Transmissions**

sql

Copy

-- Retrieve all transmissions from TRANSMISSIONS

SELECT

UPPER(name) AS "Name",

ROWID

FROM

TRANSMISSIONS;

**Query 9: Retrieve Differentials**

sql

Copy

-- Retrieve all differentials from DIFFERENTIAL

SELECT

UPPER(name) AS "Name",

ROWID

FROM

DIFFERENTIAL;

**Query 10: Search Vehicle Estimate**

sql

Copy

-- Search for vehicle estimates in VEHICLELIST based on various criteria

SELECT

vehiclelistid AS "EstimateNumber",

UPPER(vehseriescode) AS "VehSeriesCode",

UPPER(numofvehicles) AS "NumOfVehicles",

UPPER(projdesc) AS "ProjDesc",

UPPER(make) AS "Make",

UPPER(model) AS "Model",

UPPER(year) AS "Year",

UPPER(engine) AS "Engine",

UPPER(transmission) AS "Transmission",

UPPER(differential) AS "Differential",

UPPER(sopnumber) AS "SopNumber",

ROWID

FROM

VEHICLELIST

WHERE

UPPER(NVL(vehseriescode, ' ')) LIKE '%' || TRIM(UPPER(vi\_veh\_series)) || '%'

AND UPPER(NVL(projdesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_proj\_desc)) || '%'

AND UPPER(NVL(make, ' ')) LIKE '%' || TRIM(UPPER(vi\_make)) || '%'

AND UPPER(NVL(model, ' ')) LIKE '%' || TRIM(UPPER(vi\_model)) || '%'

AND UPPER(NVL(year, ' ')) LIKE '%' || TRIM(UPPER(vi\_year)) || '%'

AND UPPER(NVL(engine, ' ')) LIKE '%' || TRIM(UPPER(vi\_engine)) || '%'

AND UPPER(NVL(transmission, ' ')) LIKE '%' || TRIM(UPPER(vi\_trans)) || '%'

AND UPPER(NVL(differential, ' ')) LIKE '%' || TRIM(UPPER(vi\_diff)) || '%'

ORDER BY vehiclelistid ASC;

**Query 11: Retrieve RB Estimates**

sql

Copy

-- Retrieve all RB estimates from RBMASTERLIST

SELECT

UPPER(mmsstockcode) AS "MMSStockCode",

UPPER(rebuiltstocknum) AS "RebuiltStockNum",

UPPER(jobnumber) AS "JobNumber",

UPPER(keyword) AS "Keyword",

UPPER(detaileddesc) AS "DetailedDesc",

UPPER(corecharge) AS "CoreCharge",

UPPER(corepartnum) AS "CorePartNum",

UPPER(sopnumber) AS "SopNumber",

UPPER(buynewcost) AS "BuyNewCost",

UPPER(remancost) AS "RemanCost",

UPPER(externalcost) AS "ExternalCost",

active AS "Active",

ROWID

FROM

RBMASTERLIST;

**Query 12: Search MB**

sql

Copy

-- Search for MB parts in MBLIST based on various criteria

SELECT

UPPER(mbnumber) AS "MBNumber",

UPPER(keyword) AS "Keyword",

UPPER(description) AS "Description",

UPPER(bustype) AS "BusType",

UPPER(numbuscomp) AS "NumBusComp",

UPPER(vendorpartnum) AS "VendorPartNum",

UPPER(buycode) AS "BuyCode",

UPPER(rebuildcode) AS "RebuildCode",

UPPER(corecode) AS "CoreCode",

UPPER(annualusage) AS "AnnualUsage",

UPPER(benchnumber) AS "BenchNumber",

UPPER(sampleprovided) AS "SampleProvided",

UPPER(recommendation) AS "Recommendation",

UPPER(remancost) AS "RemanCost",

UPPER(externalcost) AS "ExternalCost",

UPPER(buynewcost) AS "BuyNewCost",

UPPER(internalcost) AS "InternalCost",

UPPER(sopnumber) AS "SopNumber",

UPPER(corecost) AS "CoreCost",

UPPER(jobnumber) AS "JobNumber",

dateopened AS "DateOpened",

ROWID

FROM

MBLIST

WHERE

UPPER(NVL(mbnumber, ' ')) LIKE '%' || TRIM(UPPER(vi\_mb\_num)) || '%'

AND UPPER(NVL(rebuildcode, ' ')) LIKE '%' || TRIM(UPPER(vi\_rebuild\_code)) || '%'

AND UPPER(NVL(keyword, ' ')) LIKE '%' || TRIM(UPPER(vi\_keyword)) || '%'

ORDER BY mbnumber DESC;

**Query 13: Retrieve MB**

sql

Copy

-- Retrieve all MB parts from MBLIST

SELECT

UPPER(mbnumber) AS "MBNumber",

UPPER(keyword) AS "Keyword",

UPPER(description) AS "Description",

UPPER(bustype) AS "BusType",

UPPER(numbuscomp) AS "NumBusComp",

UPPER(vendorpartnum) AS "VendorPartNum",

UPPER(buycode) AS "BuyCode",

UPPER(rebuildcode) AS "RebuildCode",

UPPER(corecode) AS "CoreCode",

UPPER(annualusage) AS "AnnualUsage",

UPPER(benchnumber) AS "BenchNumber",

UPPER(sampleprovided) AS "SampleProvided",

UPPER(recommendation) AS "Recommendation",

UPPER(remancost) AS "RemanCost",

UPPER(externalcost) AS "ExternalCost",

UPPER(buynewcost) AS "BuyNewCost",

UPPER(internalcost) AS "InternalCost",

UPPER(sopnumber) AS "SopNumber",

UPPER(corecost) AS "CoreCost",

UPPER(jobnumber) AS "JobNumber",

NVL(readmanualnewprice, 0) AS "ManualPrice",

ROWID

FROM

MBLIST;

**Query 14: Retrieve RB**

sql

Copy

-- Retrieve all RB parts from RBMASTERLIST

SELECT

UPPER(rebuiltstocknum) AS "RebuiltStockNum",

mmsstockcode,

keyword,

detaileddesc,

NVL(sopnumber, ' ') AS "SopNumber",

NVL(jobnumber, ' ') AS "JobNumber",

corepartnum,

corecharge,

estimatedcost,

datemodified,

active,

ROWID

FROM

RBMASTERLIST;

**Query 15: Retrieve MB Labour**

sql

Copy

-- Retrieve labour details for MB parts from EMPLOYEELABOUR

SELECT

EMPLOYEELABOUR.labourdefn AS "LabourType",

EMPLOYEELABOUR.costcentre AS "CC",

EMPLOYEELABOUR.task AS "Task",

REPLACE(EMPLOYEELABOUR.usage, ',', '.') AS "Usage",

EMPLOYEELABOUR.hrsreqd AS "Time",

EMPLOYEELABOUR.adjhrs AS "WrenchTime",

EMPLOYEELABOUR.daterevised AS "DateRevised",

EMPLOYEELABOUR.timeaddition AS "TimeAddition",

CESSETTINGS.value AS "Value",

EMPLOYEELABOUR.ROWID

FROM

EMPLOYEELABOUR,

(SELECT

CESSETTINGS.value

FROM

CESSETTINGS

WHERE

CESSETTINGS.parameter = 'NON\_PRODUCTIVE\_ADJUSTMENT') CESSETTINGS

WHERE

EMPLOYEELABOUR.linknumber = vi\_idnum

AND EMPLOYEELABOUR.typeid = vi\_type

ORDER BY

EMPLOYEELABOUR.costcentre;

**Query 16: Retrieve Labour With Totals**

sql

Copy

-- Retrieve labour details with totals from EMPLOYEELABOUR, EMPLOYEESALARIES, and EMPLCLASS

SELECT

EMPLOYEELABOUR.labourdefn AS "LabourType",

EMPLOYEELABOUR.costcentre AS "CC",

EMPLOYEELABOUR.task AS "Task",

REPLACE(EMPLOYEELABOUR.usage, ',', '.') AS "Usage",

EMPLOYEELABOUR.hrsreqd AS "Time",

EMPLOYEELABOUR.adjhrs AS "WrenchTime",

EMPLOYEESALARIES.rateperhour AS "RatePerHour"

FROM

EMPLOYEELABOUR,

EMPLOYEESALARIES,

EMPLCLASS

WHERE

EMPLOYEELABOUR.labourdefn = UPPER(EMPLCLASS.labourdefn)

AND EMPLOYEESALARIES.wagegroup = EMPLCLASS.wagegroup

AND EMPLOYEELABOUR.linknumber = vi\_idnum

AND EMPLOYEELABOUR.typeid = vi\_type;

**Query 17: Retrieve Users**

sql

Copy

-- Retrieve user details from USERS

SELECT

pswdchangedate AS "PswdChangeDate",

UPPER(accesslevel) AS "AccessLevel",

UPPER(previousaccesslevel) AS "PreviousAccessLevel",

datechangedback AS "DateChangedBack",

useoverheadfringe AS "UseOverheadFringe",

ROWID

FROM

USERS

WHERE

TRIM(UPPER(username)) = TRIM(UPPER(vi\_username))

AND TRIM(UPPER(password)) = TRIM(UPPER(vi\_password));

**Query 18: Retrieve Task Description**

sql

Copy

-- Retrieve task descriptions from LABOURTASKDESCRIPTIONS

SELECT

taskid,

UPPER(taskdescription) AS "TaskDescription"

FROM

LABOURTASKDESCRIPTIONS

ORDER BY

taskdescription ASC;

**Query 19: Retrieve Labour Position**

sql

Copy

-- Retrieve labour positions from EMPLCLASS

SELECT

UPPER(labourdefn) AS "LabourDefn"

FROM

EMPLCLASS

ORDER BY

labourdefn;

**Query 20: Retrieve Cost Centres**

sql

Copy

-- Retrieve cost centres from COSTCENTRES

SELECT

UPPER(costcentre) AS "CostCentre"

FROM

COSTCENTRES

ORDER BY

costcentre;

**Query 21: Retrieve MB All Labour**

sql

Copy

-- Retrieve all labour details for MB parts from EMPLOYEELABOUR

SELECT

UPPER(hrsreqd) AS "HrsReqd",

UPPER(usage) AS "Usage",

UPPER(task) AS "Task",

UPPER(costcentre) AS "CostCentre",

UPPER(labourdefn) AS "LabourDefn",

ROWID

FROM

EMPLOYEELABOUR

WHERE

ROWID = vi\_RowID

ORDER BY

costcentre;

**Query 22: Retrieve MB CC Lab**

sql

Copy

-- Retrieve labour details for MB parts by cost centre from EMPLOYEELABOUR, EMPLOYEESALARIES, and EMPLCLASS

SELECT

temptable.labourdefn AS "LabourType",

temptable.costcentre AS "CC",

temptable.wagegroup AS "WageGroup",

temptable.rateperhour AS "RatePerHour",

SUM(totalcost) AS "TotalCost",

SUM(timeaddition) AS "TimeAddition"

FROM

(SELECT

EMPLOYEELABOUR.labourdefn,

EMPLOYEELABOUR.costcentre,

B.wagegroup,

B.rateperhour,

ROUND((EMPLOYEELABOUR.timeaddition \* B.rateperhour \* (EMPLOYEELABOUR.usage / 100)), 2) AS TotalCost,

EMPLOYEELABOUR.timeaddition

FROM

EMPLOYEELABOUR,

EMPLOYEESALARIES B,

EMPLCLASS C

WHERE

EMPLOYEELABOUR.labourdefn = UPPER(C.labourdefn)

AND B.wagegroup = C.wagegroup

AND EMPLOYEELABOUR.linknumber = vi\_mb\_num

AND EMPLOYEELABOUR.typeid = vi\_typeid) tempTable

GROUP BY

tempTable.labourdefn,

tempTable.costcentre,

tempTable.wagegroup,

tempTable.rateperhour

ORDER BY

costcentre;

**Query 23: Retrieve NSCP Used**

sql

Copy

-- Retrieve NSC parts used from NSCPARTSUSED and NONSTOCKCODEDPARTS

SELECT

UPPER(NONSTOCKCODEDPARTS.origsuppliernum) AS "SupplierNumber",

UPPER(NONSTOCKCODEDPARTS.origsuppliername) AS "SupplierName",

NONSTOCKCODEDPARTS.oempartcost AS "UnitCost",

UPPER(NSCPARTSUSED.qtyreqd) AS "Qty",

REPLACE(NSCPARTSUSED.percentusage, ',', '.') AS "PercentUsage",

REPLACE(FINALCOST(NSCPARTSUSED.percentusage, NSCPARTSUSED.qtyreqd, NSCPARTSUSED.corecost, 0), ',', '.') AS "CoreCost",

REPLACE(FINALCOST(NSCPARTSUSED.percentusage, NSCPARTSUSED.qtyreqd, NONSTOCKCODEDPARTS.oempartcost, 0), ',', '.') AS "TotalCost",

UPPER(NSCPARTSUSED.costcentre) AS "CC",

NVL(UPPER(NONSTOCKCODEDPARTS.detaileddesc), '') AS "Description",

NVL(UPPER(NONSTOCKCODEDPARTS.keyword), '') AS "Keyword",

UPPER(NSCPARTSUSED.id) AS "ID",

UPPER(NSCPARTSUSED.lastmodifieddate) AS "LastModifiedDate",

NSCPARTSUSED.ROWID

FROM

NSCPARTSUSED,

NONSTOCKCODEDPARTS

WHERE

NSCPARTSUSED.linkcode = vi\_idnum

AND NSCPARTSUSED.linktype = vi\_linktype

AND NONSTOCKCODEDPARTS.id (+) = NSCPARTSUSED.id

ORDER BY

NONSTOCKCODEDPARTS.origsuppliernum;

**Query 24: Retrieve SCP Used**

sql

Copy

-- Retrieve SC parts used from SCPARTSUSED and STOCKCODEDPARTS

SELECT

UPPER(SCPARTSUSED.mmsstockcode) AS "StockCode",

NVL(SCPARTSUSED.mmscost, 0) AS "UnitCost",

UPPER(SCPARTSUSED.qtyreqd) AS "Qty",

REPLACE(SCPARTSUSED.percentusage, ',', '.') AS "PercentUsage",

REPLACE(FINALCOST(SCPARTSUSED.percentusage, SCPARTSUSED.qtyreqd, SCPARTSUSED.corecost, 0), ',', '.') AS "CoreCost",

REPLACE(FINALCOST(SCPARTSUSED.percentusage, SCPARTSUSED.qtyreqd, SCPARTSUSED.mmscost, 0), ',', '.') AS "TotalCost",

UPPER(SCPARTSUSED.costcentre) AS "CC",

NVL(UPPER(STOCKCODEDPARTS.detaileddesc), '') AS "Description",

NVL(UPPER(STOCKCODEDPARTS.parttype), '') AS "Keyword",

SCPARTSUSED.oemcost AS "OEMCost",

SCPARTSUSED.mmsrebuiltcode AS "MMSRebuiltCode",

SCPARTSUSED.rebpartcost AS "RebPartCost",

UPPER(SCPARTSUSED.rebuiltpart) AS "RebuiltPart",

SCPARTSUSED.ROWID

FROM

SCPARTSUSED,

STOCKCODEDPARTS

WHERE

SCPARTSUSED.linkcode = vi\_linkcode

AND SCPARTSUSED.rebuiltpart = vi\_rebuilt\_part\_choice

AND SCPARTSUSED.linktype = vi\_linktype

AND STOCKCODEDPARTS.mmsstockcode (+) = SCPARTSUSED.mmsstockcode

ORDER BY

SCPARTSUSED.mmsstockcode;

**Query 25: Retrieve RBP Used**

sql

Copy

-- Retrieve RB parts used from SCPARTSUSED, RBMASTERLIST, and TEMPMATERIALSLIST

SELECT

UPPER(SCPARTSUSED.mmsrebuiltcode) AS "RebuiltNumber",

NVL(A.RBUNITCOST, 0) AS "UnitCost",

UPPER(SCPARTSUSED.costcentre) AS "CC",

UPPER(SCPARTSUSED.qtyreqd) AS "Qty",

REPLACE(SCPARTSUSED.percentusage, ',', '.') AS "PercentUsage",

SCPARTSUSED.corecost AS "CoreCost",

NVL(A.RBCOST, 0) AS "TotalCost",

UPPER(RBMASTERLIST.detaileddesc) AS "Description",

UPPER(RBMASTERLIST.keyword) AS "Keyword",

SCPARTSUSED.ROWID

FROM

SCPARTSUSED,

RBMASTERLIST,

(SELECT

ROUND(SUM(totalcost), 2) AS RBCOST,

ROUND(SUM(totalunitcost), 2) AS RBUNITCOST,

rbreference

FROM

TEMPMATERIALSLIST

GROUP BY

rbreference) A

WHERE

SCPARTSUSED.linkcode = vi\_idnum

AND SCPARTSUSED.rebuiltpart = 'Y'

AND SCPARTSUSED.linktype = vi\_linktype

AND RBMASTERLIST.rebuiltstocknum (+) = SCPARTSUSED.mmsrebuiltcode

AND rbreference (+) = SCPARTSUSED.mmsrebuiltcode

ORDER BY

SCPARTSUSED.mmsrebuiltcode;

**Query 26: Retrieve RBP Used Distinct**

sql

Copy

-- Retrieve distinct RB parts used from SCPARTSUSED

SELECT

DISTINCT(mmsrebuiltcode) AS "RebuiltNumber"

FROM

SCPARTSUSED

WHERE

SCPARTSUSED.rebuiltpart = 'Y';

**Query 27: Retrieve RBP Used Links**

sql

Copy

-- Retrieve RB parts used links from SCPARTSUSED

SELECT

SCPARTSUSED.mmsrebuiltcode AS "RebuiltNumber",

SCPARTSUSED.linkcode AS "LinkCode"

FROM

SCPARTSUSED

WHERE

SCPARTSUSED.rebuiltpart = 'Y'

AND SCPARTSUSED.linktype = 'RB';

**Query 28: Retrieve RBP To Propagate RB**

sql

Copy

-- Retrieve RB parts to propagate RB from SCPARTSUSED

SELECT

UPPER(SCPARTSUSED.linkcode) AS "RebuiltNumber"

FROM

SCPARTSUSED

WHERE

SCPARTSUSED.rebuiltpart = 'Y'

AND SCPARTSUSED.linktype = 'RB'

AND SCPARTSUSED.mmsrebuiltcode = vi\_idnum;

**Query 29: Retrieve RBP To Propagate NSC**

sql

Copy

-- Retrieve RB parts to propagate NSC from NSCPARTSUSED

SELECT

UPPER(NSCPARTSUSED.linkcode) AS "RebuiltNumber"

FROM

NSCPARTSUSED

WHERE

NSCPARTSUSED.linktype = 'RB'

AND NSCPARTSUSED.id = vi\_idnum;

**Query 30: Retrieve RBP To Propagate SAL**

sql

Copy

-- Retrieve RB parts to propagate SAL from EMPLOYEELABOUR and EMPLCLASS

SELECT

UPPER(EMPLOYEELABOUR.linknumber) AS "RebuiltNumber"

FROM

EMPLOYEELABOUR,

EMPLCLASS

WHERE

EMPLOYEELABOUR.typeid = 'RB'

AND EMPLOYEELABOUR.labourdefn = UPPER(EMPLCLASS.labourdefn)

AND EMPLCLASS.wagegroup = vi\_idnum;

**Query 31: Retrieve RBP To Propagate LAB**

sql

Copy

-- Retrieve RB parts to propagate LAB from EMPLOYEELABOUR

SELECT

UPPER(EMPLOYEELABOUR.linknumber) AS "RebuiltNumber"

FROM

EMPLOYEELABOUR

WHERE

EMPLOYEELABOUR.typeid = 'RB'

AND EMPLOYEELABOUR.labourdefn = UPPER(vi\_idnum);

**Query 32: Retrieve Kit To Propagate NSC**

sql

Copy

-- Retrieve kits to propagate NSC from NSCPARTSUSED

SELECT

UPPER(NSCPARTSUSED.linkcode) AS "PartNumber"

FROM

NSCPARTSUSED

WHERE

NSCPARTSUSED.linktype = 'KIT'

AND NSCPARTSUSED.id = vi\_idnum;

**Query 33: Retrieve Est To Propagate RB**

sql

Copy

-- Retrieve estimates to propagate RB from SCPARTSUSED

SELECT

UPPER(SCPARTSUSED.linkcode) AS "LinkCode",

UPPER(SCPARTSUSED.linktype) AS "LinkType"

FROM

SCPARTSUSED

WHERE

SCPARTSUSED.rebuiltpart = 'Y'

AND SCPARTSUSED.mmsrebuiltcode = vi\_idnum;

**Query 34: Retrieve Kit Total Cost**

sql

Copy

-- Retrieve total cost for kits from NSCPARTSUSED and SCPARTSUSED

SELECT

NVL(SUM(FINALCOST(NSCPARTSUSED.percentusage, NSCPARTSUSED.qtyreqd, NSCPARTSUSED.cost, 0) - NSCPARTSUSED.corecost), 0) AS "NSCPTotal",

NVL(SUM(FINALCOST(SCPARTSUSED.percentusage, SCPARTSUSED.qtyreqd, SCPARTSUSED.mmscost, 0) - SCPARTSUSED.corecost), 0) AS "SCPTotal"

FROM

NSCPARTSUSED,

SCPARTSUSED

WHERE

NSCPARTSUSED.linkcode = vi\_idnum

AND NSCPARTSUSED.linktype = 'KIT'

AND SCPARTSUSED.linkcode = vi\_idnum

AND SCPARTSUSED.linktype = 'KIT'

AND SCPARTSUSED.rebuiltpart = 'N';

**Query 35: Retrieve NSC Parts Used By Supplier**

sql

Copy

-- Retrieve NSC parts used by supplier from NSCPARTSUSED

SELECT

NSCPARTSUSED.linkcode AS "LinkCode",

NSCPARTSUSED.linktype AS "LinkType"

FROM

NSCPARTSUSED

WHERE

NSCPARTSUSED.origsuppnum = vi\_orig\_supp\_num

AND NSCPARTSUSED.origsuppliername = vi\_orig\_supplier\_name;

**Query 36: Retrieve Bus From Kits Used**

sql

Copy

-- Retrieve buses from kits used from KITSUSED

SELECT

linkcode

FROM

KITSUSED

WHERE

partnumber = vi\_kit\_number;

**Query 37: Retrieve Audit 1**

sql

Copy

-- Retrieve audit history for all estimates from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

ESTIMATETYPE IN ('MB', 'RB', 'BUS', 'KIT', 'NSCP')

ORDER BY

datechanged DESC;

**Query 38: Retrieve Audit 2**

sql

Copy

-- Retrieve audit history for all estimates within a date range from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

datechanged >= TO\_DATE(vi\_date\_from)

AND datechanged <= TO\_DATE(vi\_date\_to)

AND ESTIMATETYPE IN ('MB', 'RB', 'BUS', 'KIT', 'NSCP')

ORDER BY

datechanged DESC;

**Query 39: Retrieve Audit 3**

sql

Copy

-- Retrieve audit history for a specific estimate type from AUDITMODIFYHISTORY

SELECT DISTINCT

estimatenumber

FROM

AUDITMODIFYHISTORY

WHERE

estimatetype = vi\_estimate\_type;

**Query 40: Retrieve Audit 4**

sql

Copy

-- Retrieve audit history for a specific user from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

username = vi\_user\_name

AND ESTIMATETYPE IN ('MB', 'RB', 'BUS', 'KIT', 'NSCP')

ORDER BY

datechanged DESC;

**Query 41: Retrieve Audit 5**

sql

Copy

-- Retrieve audit history for a specific user within a date range from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

username = vi\_user\_name

AND datechanged >= TO\_DATE(vi\_date\_from)

AND datechanged <= TO\_DATE(vi\_date\_to)

AND ESTIMATETYPE IN ('MB', 'RB', 'BUS', 'KIT', 'NSCP')

ORDER BY

datechanged DESC;

**Query 42: Retrieve Audit 6**

sql

Copy

-- Retrieve audit history for a specific estimate number and type from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

estimatenumber = vi\_estimate\_number

AND estimatetype = vi\_estimate\_type

AND ESTIMATETYPE IN ('MB', 'RB', 'BUS', 'KIT', 'NSCP')

ORDER BY

datechanged DESC;

**Query 43: Retrieve Audit 7**

sql

Copy

-- Retrieve audit history for a specific estimate number and type within a date range from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

estimatenumber = vi\_estimate\_number

AND estimatetype = vi\_estimate\_type

AND datechanged >= TO\_DATE(vi\_date\_from)

AND datechanged <= TO\_DATE(vi\_date\_to)

AND ESTIMATETYPE IN ('MB', 'RB', 'BUS', 'KIT', 'NSCP')

ORDER BY

datechanged DESC;

**Query 44: Retrieve Audit 8**

sql

Copy

-- Retrieve audit history for a specific user, estimate number, and type from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

username = vi\_user\_name

AND estimatenumber = vi\_estimate\_number

AND estimatetype = vi\_estimate\_type

AND ESTIMATETYPE IN ('MB', 'RB', 'BUS', 'KIT', 'NSCP')

ORDER BY

datechanged DESC;

**Query 45: Retrieve Audit 9**

sql

Copy

-- Retrieve audit history for a specific user, estimate number, and type within a date range from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

username = vi\_user\_name

AND estimatenumber = vi\_estimate\_number

AND estimatetype = vi\_estimate\_type

AND datechanged >= TO\_DATE(vi\_date\_from)

AND datechanged <= TO\_DATE(vi\_date\_to)

AND ESTIMATETYPE IN ('MB', 'RB', 'BUS', 'KIT', 'NSCP')

ORDER BY

datechanged DESC;

**Query 46: Retrieve Audit 10**

sql

Copy

-- Retrieve audit history for all settings from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

ESTIMATETYPE IN ('LTD', 'LT', 'SR', 'ETD', 'MVI', 'UAA', 'CP', 'MLB', 'CS')

ORDER BY

datechanged DESC;

**Query 47: Retrieve Audit 11**

sql

Copy

-- Retrieve audit history for all settings within a date range from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

datechanged >= TO\_DATE(vi\_date\_from)

AND datechanged <= TO\_DATE(vi\_date\_to)

AND ESTIMATETYPE IN ('LTD', 'LT', 'SR', 'ETD', 'MVI', 'UAA', 'CP', 'MLB', 'CS')

ORDER BY

datechanged DESC;

**Query 48: Retrieve Audit 12**

sql

Copy

-- Retrieve audit history for a specific user and settings from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

username = vi\_user\_name

AND ESTIMATETYPE IN ('LTD', 'LT', 'SR', 'ETD', 'MVI', 'UAA', 'CP', 'MLB', 'CS')

ORDER BY

datechanged DESC;

**Query 49: Retrieve Audit 13**

sql

Copy

-- Retrieve audit history for a specific user and settings within a date range from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

username = vi\_user\_name

AND datechanged >= TO\_DATE(vi\_date\_from)

AND datechanged <= TO\_DATE(vi\_date\_to)

AND ESTIMATETYPE IN ('LTD', 'LT', 'SR', 'ETD', 'MVI', 'UAA', 'CP', 'MLB', 'CS')

ORDER BY

datechanged DESC;

**Query 50: Retrieve Audit 14**

sql

Copy

-- Retrieve audit history for a specific setting from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

estimatetype = vi\_estimate\_type

AND ESTIMATETYPE IN ('LTD', 'LT', 'SR', 'ETD', 'MVI', 'UAA', 'CP', 'MLB', 'CS')

ORDER BY

datechanged DESC;

**Query 51: Retrieve Audit 15**

sql

Copy

-- Retrieve audit history for a specific setting within a date range from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

estimatetype = vi\_estimate\_type

AND datechanged >= TO\_DATE(vi\_date\_from)

AND datechanged <= TO\_DATE(vi\_date\_to)

AND ESTIMATETYPE IN ('LTD', 'LT', 'SR', 'ETD', 'MVI', 'UAA', 'CP', 'MLB', 'CS')

ORDER BY

datechanged DESC;

**Query 52: Retrieve Audit 16**

sql

Copy

-- Retrieve audit history for a specific user and setting from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

estimatetype = vi\_estimate\_type

AND username = vi\_user\_name

AND ESTIMATETYPE IN ('LTD', 'LT', 'SR', 'ETD', 'MVI', 'UAA', 'CP', 'MLB', 'CS')

ORDER BY

datechanged DESC;

**Query 53: Retrieve Audit 17**

sql

Copy

-- Retrieve audit history for a specific user and setting within a date range from AUDITMODIFYHISTORY

SELECT

\*

FROM

AUDITMODIFYHISTORY

WHERE

estimatetype = vi\_estimate\_type

AND username = vi\_user\_name

AND datechanged >= TO\_DATE(vi\_date\_from)

AND datechanged <= TO\_DATE(vi\_date\_to)

AND ESTIMATETYPE IN ('LTD', 'LT', 'SR', 'ETD', 'MVI', 'UAA', 'CP', 'MLB', 'CS')

ORDER BY

datechanged DESC;

**Query 54: Retrieve All Labour Summary**

sql

Copy

-- Retrieve all labour summary from TEMPLABOURLIST

SELECT

LABOURTYPE AS "LabourType",

CC AS "CC",

TASK AS "Task",

USAGE AS "Usage",

DECIMALMINUTESTOHHMM(TIME) AS "Time",

DECIMALMINUTESTOHHMM(WRENCHTIME) AS "WrenchTime",

RATEPERHOUR AS "RatePerHour",

TOTALCOST AS "TotalCost",

DECIMALMINUTESTOHHMM(ROUND(EXTENDEDTIME)) AS "ExtendedTime",

ESTIMATETYPE AS "EstimateType",

ESTIMATENUMBER AS "EstimateNumber"

FROM

TEMPLABOURLIST

ORDER BY

CC ASC, TASK ASC;

**Query 55: Retrieve All Material Summary**

sql

Copy

-- Retrieve all material summary from TEMPMATERIALSLIST

SELECT

COSTCENTRE AS "CC",

SUM(TOTALCOST) AS "TotalCost",

SUM(TAXABLECOST) AS "TaxableCost"

FROM

TEMPMATERIALSLIST

GROUP BY

COSTCENTRE

ORDER BY

COSTCENTRE ASC;

**Query 56: Retrieve All RB**

sql

Copy

-- Retrieve all RB parts from ALLMATERIALSUMMARY

SELECT

ROUND(SUM(totalcost), 2) AS "RBPrice",

rbreference

FROM

ALLMATERIALSUMMARY

GROUP BY

rbreference;

**Query 57: Report Labour Sum**

sql

Copy

-- Retrieve labour summary for a specific estimate from TEMPLABOURLIST

SELECT

CC AS "CostCentre",

SUM(extendedtime) AS "LabourHrs",

SUM(TOTALCOST) AS "TotalCost"

FROM

TEMPLABOURLIST

GROUP BY

CC

ORDER BY

CC ASC;

**Query 58: Report Mat Sum**

sql

Copy

-- Retrieve material summary for a specific estimate from TEMPMATERIALSLIST

SELECT

COSTCENTRE AS "CC",

SUM(TOTALCOST) AS "TotalCost",

SUM(TAXABLECOST) AS "TaxableCost"

FROM

TEMPMATERIALSLIST

GROUP BY

COSTCENTRE

ORDER BY

COSTCENTRE ASC;

**Query 59: R Mat Sum**

sql

Copy

-- Retrieve material summary for a specific estimate from TEMPMATERIALSLIST

SELECT

COSTCENTRE AS "CC",

SUM(TOTALCOST) AS "TotalCost",

SUM(TAXABLECOST) AS "TaxableCost"

FROM

TEMPMATERIALSLIST

GROUP BY

COSTCENTRE

ORDER BY

COSTCENTRE ASC;

**Query 60: Retrieve All Labour**

sql

Copy

-- Retrieve all labour details for a specific estimate from TEMPLABOURLIST

SELECT

CC AS "CostCentre",

DECIMALMINUTESTOHHMM(SUM(extendedtime)) AS "LabourHrs",

SUM(TOTALCOST) AS "TotalCost"

FROM

TEMPLABOURLIST

GROUP BY

CC

ORDER BY

CC ASC;

**Query 61: Retrieve One SCP Used**

sql

Copy

-- Retrieve one SC part used from SCPARTSUSED

SELECT

\*

FROM

SCPARTSUSED

WHERE

ROWID = vi\_rowid;

**Query 62: Retrieve One NSCP Used**

sql

Copy

-- Retrieve one NSC part used from NSCPARTSUSED

SELECT

\*

FROM

NSCPARTSUSED

WHERE

ROWID = vi\_rowid;

**Query 63: Find Bus Description**

sql

Copy

-- Retrieve bus description from LISTOFBUSES

SELECT

UPPER(LISTOFBUSES.listid) AS "ListId",

UPPER(LISTOFBUSES.description) AS "BusDescription"

FROM

LISTOFBUSES

WHERE

description = vi\_bus\_description;

**Query 64: Get Kit Parts**

sql

Copy

-- Retrieve kit parts from TEMPKITPARTS

SELECT

NVL(UPPER(TEMPKITPARTS.parttype), ' ') AS "PartType",

NVL(UPPER(TEMPKITPARTS.suppliernumber), ' ') AS "SupplierNumber",

NVL(UPPER(TEMPKITPARTS.suppliername), ' ') AS "SupplierName",

NVL(UPPER(TEMPKITPARTS.stockcode), ' ') AS "StockCode",

NVL(UPPER(TEMPKITPARTS.costcentre), ' ') AS "CC",

UPPER(TEMPKITPARTS.qty) AS "Qty",

REPLACE(TEMPKITPARTS.percentusage, ',', '.') AS "PercentUsage",

TEMPKITPARTS.unitcost AS "UnitCost",

REPLACE(FINALCOST(TEMPKITPARTS.percentusage, TEMPKITPARTS.qty, TEMPKITPARTS.corecost, 0), ',', '.') AS "CoreCost",

REPLACE(FINALCOST(TEMPKITPARTS.percentusage, TEMPKITPARTS.qty, TEMPKITPARTS.cost, 0), ',', '.') AS "TotalCost",

NVL(UPPER(TEMPKITPARTS.description), ' ') AS "Description",

NVL(UPPER(TEMPKITPARTS.keyword), ' ') AS "Keyword",

TEMPKITPARTS.lastmodifieddate AS "LastModifiedDate",

TEMPKITPARTS.rowid AS "RowId"

FROM

TEMPKITPARTS

ORDER BY

TEMPKITPARTS.suppliernumber,

TEMPKITPARTS.stockcode;

**Query 65: Get Sub Kits**

sql

Copy

-- Retrieve sub kits from SUBKITS

SELECT

UPPER(SUBKITS.subkitletter) AS "RefNumber",

UPPER(SUBKITS.stockcode) AS "TTCCODE",

UPPER(SUBKITS.suppliername) AS "OEM",

SUBKITS.qty AS "Qty",

UPPER(SUBKITS.suppliernumber) AS "OEMNumber",

UPPER(SUBKITS.description) AS "Description",

UPPER(SUBKITS.kitpartnumber) AS "KitPartNumber",

UPPER(SUBKITS.parttype) AS "PartType",

SUBKITS.sequence AS "Sequence"

FROM

SUBKITS

WHERE

SUBKITS.kitpartnumber = vi\_kit\_part\_number

ORDER BY

SUBKITS.sequence;

**Query 66: Get All From Sub Kits**

sql

Copy

-- Retrieve all sub kits from SUBKITS

SELECT

UPPER(SUBKITS.subkitletter) AS "RefNumber",

UPPER(SUBKITS.stockcode) AS "TTCCODE",

UPPER(SUBKITS.suppliername) AS "OEM",

SUBKITS.qty AS "Qty",

UPPER(SUBKITS.suppliernumber) AS "OEMNumber",

UPPER(SUBKITS.description) AS "Description",

UPPER(SUBKITS.kitpartnumber) AS "KitPartNumber",

UPPER(SUBKITS.parttype) AS "PartType",

SUBKITS.sequence AS "Sequence"

FROM

SUBKITS

ORDER BY

SUBKITS.sequence;

**Query 67: Get NSCP Clean 1**

sql

Copy

-- Retrieve NSC parts used with NULL orig\_supp\_num from NSCPARTSUSED

SELECT

\*

FROM

NSCPARTSUSED

WHERE

ORIGSUPPNUM IS NULL;

**Query 68: Get NSCP Clean 2**

sql

Copy

-- Retrieve NSC parts used with specific orig\_supp\_num from NSCPARTSUSED

SELECT

\*

FROM

NSCPARTSUSED

WHERE

origsuppnum = vi\_orig\_supp\_num;

**Query 69: Get NSCP Clean 3**

sql

Copy

-- Retrieve NSC parts used with NULL orig\_supplier\_name from NSCPARTSUSED

SELECT

\*

FROM

NSCPARTSUSED

WHERE

ORIGSUPPLIERNAME IS NULL;

**Query 70: Get NSCP Clean 4**

sql

Copy

-- Retrieve NSC parts used with specific orig\_supplier\_name from NSCPARTSUSED

SELECT

\*

FROM

NSCPARTSUSED

WHERE

origsuppliername = vi\_orig\_supplier\_name;

**Query 71: Get NSCP Clean 5**

sql

Copy

-- Retrieve all NSC parts used from NSCPARTSUSED

SELECT

\*

FROM

NSCPARTSUSED

ORDER BY

ROWID;

**Query 72: Get NSCP Clean 6**

sql

Copy

-- Retrieve NSC parts from NONSTOCKCODEDPARTSTEMP with specific id

SELECT

\*

FROM

NONSTOCKCODEDPARTSTEMP

WHERE

id = vi\_id;

**Query 73: Get NSCP Clean 7**

sql

Copy

-- Retrieve all NSC parts from NONSTOCKCODEDPARTS

SELECT

\*

FROM

NONSTOCKCODEDPARTS;

**Query 74: Get NSCP Clean 8**

sql

Copy

-- Retrieve NSC parts from NONSTOCKCODEDPARTSTEMP with specific orig\_supplier\_num and orig\_supplier\_name

SELECT

\*

FROM

NONSTOCKCODEDPARTSTEMP

WHERE

origsuppliernum = vi\_orig\_supplier\_num

AND origsuppliername = vi\_orig\_supplier\_name;

**Query 75: Get NSCP Clean 9**

sql

Copy

-- Retrieve NSC parts used with NULL id from NSCPARTSUSED

SELECT

\*

FROM

NSCPARTSUSED

WHERE

id IS NULL;

**Query 76: Get NSCP Clean 10**

sql

Copy

-- Retrieve NSC parts with NULL id from NONSTOCKCODEDPARTS

SELECT

\*

FROM

NONSTOCKCODEDPARTS

WHERE

id IS NULL;

**Query 77: Get NSCP Clean 11**

sql

Copy

-- Retrieve NSC parts used with NULL orig\_supp\_num from NSCPARTSUSED

SELECT

\*

FROM

NSCPARTSUSED

WHERE

ORIGSUPPNUM IS NULL;

**Query 78: Get NSCP Clean 12**

sql

Copy

-- Retrieve NSC parts with NULL orig\_supplier\_num from NONSTOCKCODEDPARTS

SELECT

\*

FROM

NONSTOCKCODEDPARTS

WHERE

ORIGSUPPLIERNUM IS NULL;

**Query 79: Get NSCP Clean 13**

sql

Copy

-- Retrieve NSC parts with NULL orig\_supplier\_name from NONSTOCKCODEDPARTS

SELECT

\*

FROM

NONSTOCKCODEDPARTS

WHERE

ORIGSUPPLIERNAME IS NULL;

**Query 80: Get NSCP Clean 14**

sql

Copy

-- Retrieve NSC parts used with specific orig\_supp\_num and orig\_supplier\_name from NSCPARTSUSED

SELECT

\*

FROM

NSCPARTSUSED

WHERE

origsuppnum = vi\_orig\_supp\_num

AND origsuppliername = vi\_orig\_supplier\_name;

**Query 81: Get NSCP Clean 15**

sql

Copy

-- Retrieve NSC parts with specific orig\_supplier\_num and orig\_supplier\_name from NONSTOCKCODEDPARTS

SELECT

\*

FROM

NONSTOCKCODEDPARTS

WHERE

origsuppliernum = vi\_orig\_supplier\_num

AND origsuppliername = vi\_orig\_supplier\_name;

**Query 82: Last Mod Date**

sql

Copy

-- Retrieve the last modification date for a specific estimate from AUDITMODIFYHISTORY

SELECT

\*

FROM

(SELECT

datechanged,

username,

estimatetype,

estimatenumber

FROM

AUDITMODIFYHISTORY

WHERE

estimatenumber = vi\_estimate\_number

AND estimatetype = vi\_estimate\_type

ORDER BY

datechanged DESC)

WHERE

ROWNUM = 1;

**Query 83: Search RB Part Bus Series**

sql

Copy

-- Search for RB parts by bus series from RBMASTERLIST and RBLISTOFBUSES

IF (vi\_search\_option = 1) THEN

SELECT DISTINCT

UPPER(RBMASTERLIST.rebuiltstocknum) AS "RebuiltNumber",

UPPER(RBMASTERLIST.mmsstockcode) AS "MMSStockCode",

UPPER(RBMASTERLIST.keyword) AS "Keyword",

UPPER(RBMASTERLIST.detaileddesc) AS "Description",

RBMASTERLIST.ROWID

FROM

RBMASTERLIST,

RBLISTOFBUSES

WHERE

RBMASTERLIST.rebuiltstocknum = RBLISTOFBUSES.rebuiltstocknum

AND UPPER(NVL(RBMASTERLIST.rebuiltstocknum, ' ')) LIKE '%' || TRIM(vi\_rebuilt\_num) || '%'

AND UPPER(NVL(RBMASTERLIST.detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND UPPER(NVL(RBMASTERLIST.keyword, ' ')) LIKE '%' || TRIM(UPPER(vi\_keyword)) || '%';

ELSIF (vi\_search\_option = 2) THEN

SELECT DISTINCT

UPPER(RBMASTERLIST.rebuiltstocknum) AS "RebuiltNumber",

UPPER(RBMASTERLIST.mmsstockcode) AS "MMSStockCode",

UPPER(RBMASTERLIST.keyword) AS "Keyword",

UPPER(RBMASTERLIST.detaileddesc) AS "Description",

RBMASTERLIST.ROWID

FROM

RBMASTERLIST,

RBLISTOFBUSES

WHERE

RBMASTERLIST.rebuiltstocknum NOT IN

(SELECT RBLISTOFBUSES.rebuiltstocknum FROM RBLISTOFBUSES)

AND NVL(RBMASTERLIST.rebuiltstocknum, ' ') LIKE '%' || TRIM(vi\_rebuilt\_num) || '%'

AND UPPER(NVL(RBMASTERLIST.detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND UPPER(NVL(RBMASTERLIST.keyword, ' ')) LIKE '%' || TRIM(UPPER(vi\_keyword)) || '%';

ELSIF (vi\_search\_option = 3) THEN

SELECT

UPPER(RBMASTERLIST.rebuiltstocknum) AS "RebuiltNumber",

UPPER(RBMASTERLIST.mmsstockcode) AS "MMSStockCode",

UPPER(RBMASTERLIST.keyword) AS "Keyword",

UPPER(RBMASTERLIST.detaileddesc) AS "Description",

RBMASTERLIST.ROWID

FROM

RBMASTERLIST,

RBLISTOFBUSES

WHERE

RBLISTOFBUSES.rebuiltstocknum = RBMASTERLIST.rebuiltstocknum

AND RBLISTOFBUSES.listid = vi\_bus\_series\_id

AND NVL(RBMASTERLIST.rebuiltstocknum, ' ') LIKE '%' || TRIM(vi\_rebuilt\_num) || '%'

AND UPPER(NVL(RBMASTERLIST.detaileddesc, ' ')) LIKE '%' || TRIM(UPPER(vi\_description)) || '%'

AND UPPER(NVL(RBMASTERLIST.keyword, ' ')) LIKE '%' || TRIM(UPPER(vi\_keyword)) || '%';

END IF;

**Query 84: Retrieve NSCP Used Copy**

sql

Copy

-- Retrieve NSC parts used copy from NSCPARTSUSED and NONSTOCKCODEDPARTS

SELECT

UPPER(NONSTOCKCODEDPARTS.origsuppliername) AS "SupplierName",

UPPER(NONSTOCKCODEDPARTS.origsuppliernum) AS "SupplierNumber",

UPPER(NSCPARTSUSED.costcentre) AS "CC",

UPPER(NSCPARTSUSED.qtyreqd) AS "Qty",

REPLACE(NSCPARTSUSED.percentusage, ',', '.') AS "PercentUsage",

NONSTOCKCODEDPARTS.oempartcost AS "UnitCost",

REPLACE(FINALCOST(NSCPARTSUSED.percentusage, NSCPARTSUSED.qtyreqd, NSCPARTSUSED.corecost, 0), ',', '.') AS "CoreCost",

UPPER(NSCPARTSUSED.id) AS "ID",

REPLACE(FINALCOST(NSCPARTSUSED.percentusage, NSCPARTSUSED.qtyreqd, NSCPARTSUSED.cost, 0), ',', '.') AS "TotalCost",

NVL(UPPER(NONSTOCKCODEDPARTS.detaileddesc), '') AS "Description",

NVL(UPPER(NONSTOCKCODEDPARTS.keyword), '') AS "Keyword",

UPPER(NSCPARTSUSED.lastmodifieddate) AS "LastModifiedDate",

NSCPARTSUSED.ROWID

FROM

NSCPARTSUSED,

NONSTOCKCODEDPARTS

WHERE

NSCPARTSUSED.linkcode = vi\_idnum

AND NSCPARTSUSED.linktype = vi\_linktype

AND NONSTOCKCODEDPARTS.id (+) = NSCPARTSUSED.id

ORDER BY

NSCPARTSUSED.origsuppnum;

These queries have been extracted and modified to remove underscores from table and column names. They are now ready for use in a SQL Server environment after any necessary adjustments for SQL Server syntax.