```
CL:CHGLIBL LIBL(QTEMP ComSQLQry Systools QGPL);

--+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
-+-+-+-
-- DATA CENTRIC
--+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
-+-+-+-
-- Views
--
===============================================================================================
=====
-- Quarterly Sales
-----------------------------------------------------------------------------------------------
-------
Select * from Sales;
Select * from AddressX;

Select   CustNo, Year(SalesDate) as SalesYear,
         sum(case when Quarter(SalesDate)= 1 then Amount else 0 end) as Q1,
         sum(case when Quarter(SalesDate)= 2 then Amount else 0 end) as Q2,
         sum(case when Quarter(SalesDate)= 3 then Amount else 0 end) as Q3,
         sum(case when Quarter(SalesDate)= 4 then Amount else 0 end) as Q4,
         sum(Amount) as Total
   from  Sales
   Group by CustNo, Year(SalesDate)
   -- group by grouping sets((CustNo, Year(SalesDate)), (CustNo), ())
   Order By CustNo, SalesYear
;

Drop View ComSQLQry.SalesQuart;

Commit;

Create or Replace View ComSQLQry.SalesQuart as
Select   CustNo,
         Year(SalesDate) as SalesYear,
         Cast(sum(case when Quarter(SalesDate)= 1 then Amount else 0 end) as Dec(11, 2)) as Q1,
         Cast(sum(case when Quarter(SalesDate)= 2 then Amount else 0 end) as Dec(11, 2)) as Q2,
         Cast(sum(case when Quarter(SalesDate)= 3 then Amount else 0 end) as Dec(11, 2)) as Q3,
         Cast(Sum(Case When Quarter(SalesDate)= 4 Then Amount else 0 End) as Dec(11, 2)) as Q4,
         Cast(Sum(Amount) as Dec(11, 2) as Total
  From Sales
  GRoup By CustNo, Year(SalesDate);
Commit;

Select * from SalesQuart;
-----------------------------------------------------------------------------------------------
-------
-- Quarterly Sales with Address Information
-----------------------------------------------------------------------------------------------
-------
-- Drop View ComSQLQry.SalesQCust;
-- Commit

Create or Replace View ComSQLQry.SalesQCust
    as Select a.CustNo as ACustNo, SalesYear, Q1, Q2, Q3, Q4, Total,
              b.*
          from SalesQuart a Left outer join AddressX b on a.CustNo = b.CustNo;


Select * from SalesQCust;

Commit;

Select SalesYear, ACustNo, CustName1, City, Q1, Q2, Q3, Q4, Total
from SalesQCust
```

```
Where SalesYear = 2009 and ACustNo in ('10003', '10005');

-------------------------------------------------------------------------------------------------------
-------
-- Exchange Rates: View including Webservice access
-------------------------------------------------------------------------------------------------------
-------
Drop View ComSQLQry.EXCRATEV1;
Drop View ComSQLQry.EXCRATEV2;

-- Create Or Replace View ComSQLQry.EXCRATEV1 as
 Select  a.*, Cast(Round(1,00 / ExcRate, 4) as Dec(7, 4)) RevRate
 From XMLTable(
        XMLNamespaces(DEFAULT 'http://www.ecb.int/vocabulary/2002-08-01/eurofxref',
                              'http://www.gesmes.org/xml/2002-08-01' AS "gesmes"),
        'gesmes:Envelope/Cube/Cube/Cube'
        Passing XMLParse(DOCUMENT
                   HTTP_GET('https://www.ecb.europa.eu/stats/eurofxref/eurofxref-daily.xml',
'')
                --   HTTPGetBLOB('https://www.ecb.europa.eu/stats/eurofxref/eurofxref-
daily.xml', '')
                 )
        Columns Subject   VarChar(30) Path '../../../gesmes:subject',
                Sender    VarChar(30) Path '../../../gesmes:Sender/gesmes:name',
                ExcDate   Date        Path '../@time',
                Currency  Char(3)     Path '@currency',
                ExcRate   Dec(10, 4)  Path '@rate'      ) a
;

--    HTTP_GET('https://www.ecb.europa.eu/stats/eurofxref/eurofxref-daily.xml')

Commit;

Select * from EXCRATEV1;

Drop View ComSQLQry.EXCRATEV2;

Create Or Replace View ComSQLQry.EXCRATEV2 as
Select s.*, Cast(Round(Amount * ExcRate, 4) as Dec(11, 4)) Amount_USD
   from Sales s cross join (Select ExcRate from ExcRateV1 Where Currency = 'USD') ;

Commit;

Select * from ComSQLQry.EXCRATEV2;
Stop;
-------------------------------------------------------------------------------------------------------
-------
-- View including Global Variables
-------------------------------------------------------------------------------------------------------
-------
Drop Variable COMSQLQRY.GBLCURRENCY;

-- Global Variable for Currency
Create Or Replace Variable COMSQLQRY.GBLCURRENCY
                             For System Name GBLCURRNCY
                             VARCHAR( 10) Default  'USD'  ;

Create Or Replace View ComSQLQry.EXCRATEV3 as
Select s.*, Cast(Round(Amount * ExcRate, 4) as Dec(11, 4)) Amount_Foreign_Currency,
            GblCurrency as Foreign_Currency, ExcRate
   from Sales s cross join (Select ExcRate from ExcRateV1 Where Currency = GblCurrency) ;

Commit;

Set GblCurrency = 'USD';
Set GblCurrency = 'GBP';
```

```
Set GblCurrency = 'CHF';
Set GblCurrency = 'SEK';
Set GblCurrency = 'JPY';

Select * From EXCRATEV3
Order By CustNo;



-------------------------------------------------------------------------------------------------------
-------
-- View: Recursion with Global Variables
-------------------------------------------------------------------------------------------------------
-------
Drop Variable ComSQLQry.GblDeparture;
Drop Variable ComSQLQry.GblArrival;
Drop Variable ComSQLQry.GblMaxConnects;

 Create or Replace Variable ComSQLQry.GblDeparture   VarChar(50) Default 'Frankfurt';
 Create or Replace Variable ComSQLQry.GblArrival     VarChar(50) Default 'Berlin';
 Create or Replace Variable ComSQLQry.GblMaxConnects Integer Default 3;

Commit;

-- Create View with global variables
-------------------------------------*
Select * from Flights
Order By Departure, Arrival;

Drop view ComSQLQry.FlightV03;

Create Or Replace View ComSQLQry.FlightV03
    as
    Select Connect_By_Root Departure as Departure, Arrival,
              Cast(Connect_By_Root Departure concat Sys_Connect_By_Path(Arrival, ' -> ')
                   as VarChar(1028)) as Itinerary,
              Level - 1 as NbrConnect,
              Calculate(Sys_Connect_By_Path(VarChar(Price), '+')) as Costs
          From Flights

          Where     Arrival             = GblArrival
                And Connect_By_isCycle  = 0
                and Level              <= GblMaxConnects + 1

          Start With Departure = GblDeparture
          Connect By NoCycle Prior     Arrival = Departure
                                   and Arrival <> GblDeparture;


Commit;

-- Values GblConnect;
Set GblDeparture = 'München', GblArrival = 'Hamburg', GblMaxConnects = 2;

 Select Departure, Arrival, Itinerary, Costs
 from flightV03
 Order By Costs;

 Set GblDeparture = 'München',   GblArrival = 'Hamburg', GblMaxConnects = 2;
 Set GblDeparture = 'Frankfurt', GblArrival = 'Berlin',  GblMaxConnects = 3;
 Set GblDeparture = 'Berlin',    GblArrival = 'Köln',    GblMaxConnects = 2;

 Select * from flightV03
 Order By Costs
 Limit 1;
```

```
-------------------------------------------------------------------------------------------------------
-------
-- Country Names
-------------------------------------------------------------------------------------------------------
-------
Values(HTTP_Get('https://pkgstore.datahub.io/core/country-list/latest/data/json/data.json' ,
''));

Select *
   From JSON_TABLE(HTTP_GET('https://pkgstore.datahub.io/core/country-list/latest/data/json/
data.json'
            , ''),
            '$[*]'
            COLUMNS(CODE    Varchar(5)  Path '$.Code',
                    COUNTRY Varchar(50) Path '$.Name'))  X;

Create or Replace View COMSQLQRY / COUNTRYCDE(CODE, COUNTRY)
   As Select *
         From JSON_TABLE(HTTP_GET('https://pkgstore.datahub.io/core/country-list/latest/data/
json/data.json'
            , ''),
            '$[*]'
               COLUMNS(CODE    Varchar(5)  Path '$.Code',
                       COUNTRY Varchar(50) Path '$.Name'))  X
   Rcdfmt COUNTRYCDE;

Select * from CountryCde;

Commit;

-- Join Address Table with "Webservice"
-----------------------------------------
Select CustNo, CustName1, a.Country CountryCode, Coalesce(c.Country, '') CountryDescr, ZipCode,
City
  from Addressx a Left join CountryCde c on a.Country = Code;

-------------------------------------------------------------------------------------------------------
-------
-- Country Information
-------------------------------------------------------------------------------------------------------
-------
Select *
  from JSON_TABLE(http_get('http://www.geognos.com/api/en/countries/info/DE.json', ''),
                  '$.Results'
                  Columns("Name"    VarChar(25),
                          Capital   VarChar(25)   Path '$.Capital.Name',
                          ISO2       VarChar(2)   Path '$.CountryCodes.iso2',
                          ISO3       VarChar(3)   Path '$.CountryCodes.iso3',
                          GeoPt1     Dec(7, 2)    Path '$.GeoPt[0]',
                          GeoPt2     Dec(7, 2)    Path '$.GeoPt[1]',
                          Web        VarChar(256) Path '$.CountryInfo')) x;

Create Or Replace Variable ComSQLQry/GblCountry2
          For System Name GBLCNTRY2
          VarChar(2)  Default 'DE';

Create Or Replace View ComSQLQry.Country_Info
          For System Name CntryInfWV
as
Select *
  from JSON_TABLE(http_get('http://www.geognos.com/api/en/countries/info/' concat
Trim(GblCountry2) concat '.json', ''),
                  '$.Results'
                  Columns("Name"    VarChar(25),
                          Capital   VarChar(25)   Path '$.Capital.Name',
                          ISO2       VarChar(2)   Path '$.CountryCodes.iso2',
```

```
                            ISO3       VarChar(3)   Path '$.CountryCodes.iso3',
                            GeoPt1     Dec(7, 2)    Path '$.GeoPt[0]',
                            GeoPt2     Dec(7, 2)    Path '$.GeoPt[1]',
                            Web        VarChar(256) Path '$.CountryInfo')) x;

Commit;

Set GblCountry2 = 'FR';
Set GblCountry2 = 'GB';
Set GblCountry2 = 'US';
Set GblCountry2 = 'BE';
Set GblCountry2 = 'SE';

Select * from Country_Info;
-------------------------------------------------------------------------------------------------------
-------
-- Country Information and Address Master
-------------------------------------------------------------------------------------------------------
-------
Select * From AddressX

-- Passing Information from the first table as parameters to the JSON_TABLE function
-----------------------------------------------------------------------------------------
-- 1. Only the Countries with a valid country code are selected
-- 2. Too slow because the webservice is called for each row!!!
Select CustNo, CustName1, a.Country CountryCode, Coalesce(Name, '') CountryDescr, ZipCode,
City, c.*
   from AddressX a Cross Join
   Lateral(Select *
              from JSON_TABLE(http_get('http://www.geognos.com/api/en/countries/info/' concat
Trim(Country) concat '.json', ''),
                                    '$.Results'
                               Columns(Name       VarChar(25)  Path '$.Name',
                                       Capital    VarChar(25)  Path '$.Capital.Name',
                                       ISO2       VarChar(2)   Path '$.CountryCodes.iso2',
                                       ISO3       VarChar(3)   Path '$.CountryCodes.iso3',
                                       GeoPt1     Dec(7, 2)    Path '$.GeoPt[0]',
                                       GeoPt2     Dec(7, 2)    Path '$.GeoPt[1]',
                                       Web        VarChar(256) Path '$.CountryInfo')) x) c;

-- Address Information and Web-service reworked
-----------------------------------------------
-- Create or Replace View COMSQLQRY.Address_Country_WV01
--           For System Name ADDRCTYWV
-- as
With Cty as (Select Distinct Country as CtyCode
               from Addressx
               Where Length(Trim(Country)) = 2,
      CtyInfo as (Select *
                    from Cty Cross Join
                         Lateral(Select *
                                   from JSON_TABLE(http_get('http://www.geognos.com/api/en/
countries/info/' concat Trim(CtyCode) concat '.json', ''),
                                                          '$.Results'
                                                     Columns(Name       VarChar(25)   Path
'$.Name',
                                                             Capital    VarChar(25)   Path
'$.Capital.Name',
                                                             ISO2       VarChar(2)   Path
'$.CountryCodes.iso2',
                                                             ISO3       VarChar(3)   Path
'$.CountryCodes.iso3',
                                                             GeoPt1     Dec(7, 2)    Path
'$.GeoPt[0]',
                                                             GeoPt2     Dec(7, 2)    Path
'$.GeoPt[1]',
```

```
                                                                  Web        VarChar(256) Path
'$.CountryInfo')) x) c)
Select *
  from AddressX a Left join CtyInfo c on a.Country = c.CtyCode;

Commit;

Select * from Address_Country_WV01;

--
********************************************************************************************
*****
-- SQL Routines
--
********************************************************************************************
*****
-- Stored Procedures
--
============================================================================================
=====
-- Calculating Workdays
--------------------------------------------------------------------------------------------
-------
Create Or Replace Procedure ComSQLQry.GetWrkDay_SP
                 (In  ParStrDate  Date    Default Current_Date,
                  In  ParEndDate  Date    Default Current_Date,
                  Out POutNbrDays Integer)
              Language SQL
              Specific GETWRKDSP
              Deterministic
              Called on NULL Input

              Set Option  DBGVIEW   = *SOURCE

BEGIN
   Declare LocDayOfWeek   SmallInt Default 0;
   Declare LocDate        Date     Default '0001-01-01';

   Set LocDate      = ParStrDate;
   Set LocDayOfWeek = DayOfWeek_ISO(ParStrDate);
   Set POutNbrDays  = 0;

   BegRepeat:
      Repeat If LocDayOfWeek < 6 Then Set POutNbrDays = POutNbrDays + 1;
             End If;
             If LocDayOfWeek = 7 Then Set LocDayOfWeek = 1;
             Else Set LocDayOfWeek = LocDayOfWeek + 1;
             End If;
             Set LocDate = LocDate + 1 Days;

      Until LocDate > ParEndDate
      End Repeat;

   Return POutNbrDays;
END  ;

 Comment on Specific Procedure ComSQLQry.GETWRKDSP
      is 'Determine Number of Workdays';

Commit;

Call GetWrkDay_SP(Current_Date , Date('2024-12-31') , ? );

With Calendar (rundate) as (Values(Current_date)
                            Union All
                            Select Rundate + 1 Day From Calendar
```

```
                              where Rundate < Right(Year(Current_Date), 4) concat '-12-31')
Select Count(*) -- RunDate
From Calendar
Where DayOfWeek_Iso(rundate) < 6;
--
================================================================================================
=====
-- User Defined Functions
--
================================================================================================
=====
-- Convert numeric Date
------------------------------------------------------------------------------------------------
-------
Create or Replace Function ComSQLQry.CVTDATE (
       ParDateNum Dec(8, 0) )
       Returns DATE
       Language SQL
       Specific CVTDATE
       Deterministic
       Reads SQL Data
       Called on NULL Input
       Disallow Parallel
       Set Option  DBGVIEW   = *SOURCE

BEGIN
   Declare Continue Handler For SQLEXCEPTION
   Return Date('8888-12-31') ;
   Return Date(Digits(ParDateNum) Concat '000000');
END  ;

Commit;

Select NumDate, CvtDate(NumDate) Converted_Date
from NumDate a;
------------------------------------------------------------------------------------------------
-------
-- Determine Work Days for a specific Date Range
------------------------------------------------------------------------------------------------
-------
Create Or Replace Function ComSQLQry.GetWrkDay
         (ParBegDate Date Default Current_Date,
          ParEndDate Date Default Current_Date)
       Returns Integer
       Language SQL
       Specific GetWrkDay
       Deterministic
       Reads SQL Data
       Called On Null Input

       Set Option Dbgview = *Source
 Begin
   Declare Continue Handler For SQLException Return -1;
   Return With    Calendar (RunDate) as (Values(ParBegDate)
                Union All
                   Select Rundate + 1 Day
                     From Calendar
                     where Rundate < ParEndDate)
           Select Count(*)
              From Calendar
              Where DayOfWeek_Iso(rundate) < 6;
 End;

 Commit;

 Values(GetWrkDay(Current_Date, '2024-12-31'));
```

```
 Select GetWrkDay(SalesDate, SalesDate + 2 Months) WorkDays, SalesDate
   From Sales a;

-------------------------------------------------------------------------------------------------------
-------
-- Get Monday from year / week
-------------------------------------------------------------------------------------------------------
-------
Create or Replace Function ComSQLQry.GetMondayFromYearWeek
         (ParYear     Decimal(4, 0)  Default (Year(Current_Date)),
          ParWeekISO  Decimal(2, 0)  Default (Week_ISO(Current_Date)))
       Returns Date
    Language SQL
    Specific MONYYYYMM
    Deterministic
    Modifies SQL Data
    Called On NULL Input

    Set Option DBGView = *Source

Begin
   Declare Jan4  Date  Default '0001-01-01';
   Declare Continue Handler
           For SQLException Return Date('0001-01-01');

   Set Jan4 = Date(Digits(ParYear) concat '-01-04');

   If   ParWeekISO < 0
     Or ParWeekISO > Week_ISO( Jan4 - 1 Year - 7 Days)
     Then Return Date('0001-01-01');
   End IF;

   Return Jan4 + (((ParWeekIso - 1) * 7) - DayOfWeek_ISO(Jan4) + 1) Days;
End;

Values(GetMondayFromYearWeek(2024, 2)),
      (GetMondayFromYearWeek(2024, Week_ISO(Current_Date)));

-------------------------------------------------------------------------------------------------------
-------
-- SQL UDF - Calculate
-------------------------------------------------------------------------------------------------------
-------
-- Drop Specific Function ComSQLQry.CALCUCLOB;
-- Drop Specific Function ComSQLQry.CALCULATE;

Create or Replace Function ComSQLQry.CALCULATE
          (ParToCalc VarChar(256))
       Returns Decimal(31, 9)
       Language SQL
       Specific CALCULATE
       Modifies SQL Data
       Called On NULL Input
       No External Action
       Not Fenced

       Set Option DbgView = *Source

Begin
   Declare RtnVal     Decimal(31, 9);
   Declare DynSQLStmt VarChar(350);

   Declare Continue Handler for SQLEXCEPTION
           Return -9999999999999,999999999;
```

```
    If ParToCalc is NULL or ParToCalc = '' Then Return 0;
    End If;

    Set DynSQLStmt = 'Values(' concat Trim(ParToCalc) concat ') into ?';

    Prepare DynSQL from DynSQLStmt;
    Execute DynSQL using RtnVal;
    Return RtnVal;
 End;

 Comment On  Specific Function CALCULATE
  is 'Calculate String';

Commit;

 Values(Calculate(' 3 + 7*2'), Calculate( 2**10), Calculate(2 * ((3+5) - 2**2)));
 Values(Calculate('Time(''19.00.00'') - Current_Time'));
 Values(Calculate('sin(1,5)'));

--
================================================================================================
=====
-- User Defined Table Function
--
================================================================================================
=====
-- 1. Calling user Defined Table Functions
------------------------------------------------------------------------------------------------
-------
-- 1.1. UDTF without parameters
-------------------------------
 Select * from Table(USERS()) u;;
 Select * from Table(Schemas()) x;;

-- 1.2. Using WHERE conditions with UDTFs
----------------------------------------*
 Select * from Table(USERS()) u
    Where ODOBNM like '%HAUS%';;

-- 1.3. UDTF with parameters: ListMember_Fnc (List Member)
-------------------------------------------------------------------------*
 Select * from Table(Message_File_Data('QSYS', 'QSQLMSG'));

  Select * from Table(Qsys2.Message_File_Data(Message_File_Library => 'QSYS',
                                              Message_File => 'QSQLMSG'))
    Where Message_Id like 'SQL015%';

  Select * from Table(COMDBPGM.ListMember_Fnc('QSQLPGM', 'COMDBPGM')) x
      Where     MbrType like '%RPGLE%'
            and Mbr     like '%LIST%'
;

-- 1.3. Joining UDTFs
---------------------------------*
--  Select a.MbrFile, a.MbrFileLib, a.Mbr, a.MbrDescr,
--         b.MbrFile, b.MbrFileLib, b.Mbr, b.MbrDescr
--     From           Table(COMDBPGM.ListMember_Fnc('QSQLPGM', 'COMDBPGM')) a
--          Full Join Table(COMDBPGM.ListMember_Fnc('QRPGLESRC', 'HSCOMMON10')) b
--                 on a.Mbr = b.Mbr
--        Where (   a.MbrDescr like '%File%'
--               or b.MbrDescr like '%File%');

Select a.Object_Name, a.Object_Type, a.Object_Attribute,
       b.Object_Name, b.Object_Type, b.Object_Attribute
  from Table(QSYS2.Save_FILE_Objects(Save_File => 'COMDBMOD', Save_File_Library => 'BHATRANS'))
a
```

```
       Full Join
       Table(QSYS2.Save_FILE_Objects(Save_File => 'COMSQLQRY', Save_File_Library =>
'BHATRANS')) b
       on     a.Object_Name = b.Object_Name
          and a.Object_Type = b.Object_Type
          and a.Object_Attribute = b.Object_Attribute
  Order By a.Object_Name, a.Object_Type, a.Object_Attribute,
           b.Object_Name, b.Object_Type, b.Object_Attribute
       ;

-- 1.4. Merging UDTFs
-------------------------------------*

   Select * from Table(ComDBPGM.ListMember_Fnc('QRPGLESRC', 'HSCOMMON05')) a
      Where     MbrType like '%RPGLE%'
            and Mbr     like '%LIST%'
 Union All
   Select * from Table(COMDBPGM.ListMember_Fnc('QSQLPGM', '*LIBL')) b
     Where     MbrType like '%RPGLE%'
           and Mbr     Like '%LIST%'
 Order By Mbr, MbrFileLib, MbrFile
;

-- 1.5. Joining UDTFs using Lateral
----------------------------------------*
 Select * from Table(Users()) x;

-- Select * from Table(DspObjOwn_Fnc('HAUSER')) x;
 Select * from Table(QSys2.Object_OwnerShip(User_Profile => 'HAUSER'));

--   Select ODOBTX, p.*
--    From Table(Users()) u cross join
--         Lateral(Select *
--                   From Table(DspObjOwn_Fnc(Odobnm)) x) p
--    Where     Odobnm Like 'HAUS%'
--          and ObjType in ('*DTAARA', '*BNDDIR', '*SQLUDT')
--    Order By ObjType, ObjLib, Obj
-- ;

Select ODOBTX, p.*
  From Table(Users()) u cross Join
       Lateral(Select *
                 from Table(QSys2.Object_OwnerShip(User_Profile => ODOBNM))) p
  Where     Odobnm Like 'HAUS%'
        and Object_Type in ('*DTAARA', '*BNDDIR', '*SQLUDT')
  Order By Object_Type, Object_Library, Object_Name;

-------------------------------------------------------------------------------------------------
-------
-- CL Command with OutFile
-------------------------------------------------------------------------------------------------
-------
Create Or Replace Function ComSQLQry.DspBndDir_Fnc
         (ParBnddir    VarChar(10),
          ParBnddirLib VarChar(10)  Default '*LIBL')
       Returns Table (BndDirLib  Char(10),
                      BndDir     Char(10),
                      ObjLib     Char(10),
                      ObjName    Char(10),
                      ObjType    Char(10),
                      Activation Char(10))
       Language SQL
       Specific DspBnddir
       Not Deterministic
       Modifies SQL Data
       Called On NULL Input
```

```
       Disallow Parallel
       Not Fenced

       Set Option DBGVIEW = *Source
 Begin
    -- Create OutFile
    Call QSys2.QCMDEXC(
               'DSPBNDDIR BNDDIR(' concat Trim(ParBnddirLib) concat
                                      '/'                   concat
                                      Trim(ParBndDir)       concat
                                      ') '                  concat
                        ' OUTPUT(*OUTFILE) '                concat
                        ' OUTFILE(QTEMP/TMPBNDDIR)');

     -- Return Result
     Return Select BNDRLB, BNDRNM, BNOLNM, BNOBNM, BNOBTP, BNOACT
              From TMPBNDDIR;
 End;

 Commit;

 Select *
   From Table(DspBndDir_Fnc('BXBNDDIR', 'BXOBJ'));

 Select *
   from Table(QSYS2.Binding_Directory_Info(Binding_Directory_Library => 'BXOBJ',
                                           Binding_Directory         => 'BXBNDDIR'));

-------------------------------------------------------------------------------------------
-- Materialized Query Tables (MQT)
-------------------------------------------------------------------------------------------
-- Sales per Month
------------------------
Create Or Replace Table ComSQLQry.SalesMon
       as (Select Year(Salesdate) As SalesYear,
                  Month(Salesdate) As SalesMonth,
                  Custno,
                  Sum(Amount)      As AmountYear
                From ComSQLQry.Sales
                Group By Year(SalesDate), Month(SalesDate), Custno)
   Data Initially Immediate
   Refresh Deferred
   Maintained By User
   Enable Query Optimization
   Rcdfmt Salesmon;

Commit;

Select * from SalesMon;

-- Sales per year joined with the address master table
-------------------------------------------------------
Create Or Replace Table ComSQLQry/SalesYear
       As (Select Year(Salesdate)      As Salesyear,
                   A.Custno, Custname1, Country,
                  Zipcode,   City,      Street,
                  Sum(Amount)           As Amountyear
                From    ComSQLQry.Sales S Join ComSQLQry.Addressx A
                     On S.CustNo = A.CustNo
                Group By Year(Salesdate), A.Custno, Custname1, Country,
                         Zipcode,         City,     Street)
   Data Initially Immediate
   Refresh Deferred
   Maintained By User
   Enable Query Optimization
   Rcdfmt Salesyear;
```

```
Commit;

Select * from SalesYear;

--
*************************************************************************************************
************************************
-- Check Constraints
--
*************************************************************************************************
************************************
-- Attention: Examples Only
Alter Table COMDBMOD.ZZ_ORDER_HEADER
   Add Constraint COMDBMOD.ZZORDHP_ORDER_TYPE_00001
       Check(Order_Type in ('DO', 'EX', 'UO'))
   Add Constraint COMDBMOD.ZZORDHP_DELIVERY_TERMS_00001
       Check(DELIVERY_TERMS in ('CPT', 'EXW'))
   Add Constraint COMDBMOD.ZZORDHP_ORDER_HEADER_STATUS_00001
       Check(Order_Header_Status in ('EN', 'OPN', 'CP', 'PD', 'CL', 'FIN'))       ;

Alter Table COMDBMOD.ZZ_ORDER_DETAIL
   Add Constraint COMDBMOD.ZZORDDP_DELIVERY_QUANTITY_00001
       Check(DELIVERY_QUANTITY >= 0)

   Add Constraint COMDBMOD.ZZORDDP_ORDER_POSITION_STATUS_00001
       Check(ORDER_POSITION_STATUS In ('EN', 'PD', 'CP', 'CL'))

   Add Constraint COMDBMOD.ZZORDDP_ORDER_QUANTITY_00001
       Check(ORDER_QUANTITY > 0 and ORDER_QUANTITY >= DELIVERY_QUANTITY);

--
*************************************************************************************************
************************************
-- Referential Integrities
--
*************************************************************************************************
************************************
-- Attention: Examples Only
-- 1. Order Header Table
Alter Table COMDBMOD.ZZ_ORDER_HEADER
   Add Foreign Key (OHADID)
               References ZZ_ADDRESS_MASTER (ADID)
       On Delete Restrict
       On Update Restrict;

-- 2. Order Detail Table
Alter Table COMDBMOD.ZZ_ORDER_DETAIL
   Add Foreign Key (ODITID)
               References ZZ_ITEM_MASTER (ITID)
               On Delete Restrict
               On Update Restrict;

-- Attention: Order Header without Order Positions and vice versa --> Referential Integrity
cannot implemented
Alter Table COMDBMOD.ZZ_ORDER_DETAIL
   Add Foreign Key (ODOHID)
               References COMDBMOD/ZZ_ORDER_HEADER (OHID)
               On Delete Restrict
               On Update Restrict;


--*************************************************************************************
-- Trigger
--*************************************************************************************
-- 1. Before Insert Trigger
```

```
-- Single Statement Trigger
-------------------------------------------------------------
Create or Replace Trigger COMDBPGM.BITNXTPOS
       Before Insert On COMDBPGM.ORDERDETX
       Referencing New as N
       For Each Row
       Mode DB2Row
       Set Option  Commit = *CHG,
                   DbgView = *SOURCE

  Select Coalesce(Max(x.OrderPos), 0) + 1
         Into N.OrderPos
    from COMDBPGM.OrderDetX x
    Where x.Company = n.Company and x.OrderNo = n.OrderNo;



-- 3. Add a Before Insert Trigger that automatically updates the new date column
-------------------------------------------------------------------------------------------
-- Attention: Example Only
Drop Trigger COMDBMOD.ORDER_HEADER_B4IU_DELDATE;

Create Or Replace Trigger COMDBMOD.ORDER_HEADER_B4IU_DELDATE
          Before Insert Or Update Of DELIVERY_DATE_NUM, DELIVERY_DATE
          On COMDBMOD.ZZ_ORDER_HEADER
          Referencing New as N
                      Old as O
          For Each Row
          Mode DB2ROW
          Program Name ZZOHBIU01
          Not SECURED
    Begin Atomic
          Declare LocType    VarChar(10) Default '';
          Declare Continue Handler for SQLEXCEPTION Set N.Delivery_Date = '8888-12-31';

          If    Inserting and N.Delivery_Date > '0001-01-01'
             or Updating  and O.Delivery_Date <> N.Delivery_Date
             Then Set N.Delivery_Date_Num = Dec(N.Delivery_Date, 8, 0);
          ElseIf Inserting and N.Delivery_Date_Num > 10101
             or Updating and O.Delivery_Date_Num <> N.Delivery_Date_Num
             Then Set N.Delivery_Date = Date(Digits(N.Delivery_Date_Num) concat '000000');
          ElseIf Inserting
             Then Set N.Delivery_Date     = '0001-01-01';
                  Set N.Delivery_Date_Num = 0;
          End If;
    End;

Commit;
Rollback;

-------------------------------------------------------------------------------------------------------
-----------
-- Table With Auditing Columns
-------------------------------------------------------------------------------------------------------
-----------
CREATE OR REPLACE TABLE COMSQLQRY.MYADDTABLE (
        UNQID INTEGER GENERATED ALWAYS AS IDENTITY (START WITH 100 INCREMENT BY 10
                                                    NO MINVALUE NO MAXVALUE
                                                    NO CYCLE NO ORDER
                                                    CACHE 20 ),
        INTVAL      INTEGER                 NOT NULL DEFAULT 0 ,
        CHGTIMESTP  TIMESTAMP              GENERATED ALWAYS FOR EACH ROW ON UPDATE AS ROW
CHANGE TIMESTAMP NOT NULL ,
        CHGACCTNG   VARCHAR(255) CCSID 273 GENERATED ALWAYS AS ( CURRENT CLIENT_ACCTNG ) ,
        CHGAPPNAME  VARCHAR(255) CCSID 273 GENERATED ALWAYS AS ( CURRENT CLIENT_APPLNAME ) ,
        CHGPGMID    VARCHAR(255) CCSID 273 GENERATED ALWAYS AS ( CURRENT CLIENT_PROGRAMID ) ,
```

```
        CHGUSERID   VARCHAR(255) CCSID 273 GENERATED ALWAYS AS ( CURRENT CLIENT_USERID ) ,
        CHGWRKSTN   VARCHAR(255) CCSID 273 GENERATED ALWAYS AS ( CURRENT CLIENT_WRKSTNNAME ) ,
        CHGCURRSVR  VARCHAR(18)  CCSID 273 GENERATED ALWAYS AS ( CURRENT SERVER ) ,
        CHGSESSUSR  VARCHAR(128) CCSID 273 GENERATED ALWAYS AS ( SESSION_USER ) ,
        CHGUSER     VARCHAR(18)  CCSID 273 GENERATED ALWAYS AS ( USER ) ,

        CHGJOB      VARCHAR(28)  CCSID 273 GENERATED ALWAYS AS ( QSYS2.JOB_NAME ) ,
        CHGSVRMODE  VARCHAR(28)  CCSID 273 GENERATED ALWAYS AS ( QSYS2.SERVER_MODE_JOB_NAME ) ,
        CHGHOST     VARCHAR(255) CCSID 273 GENERATED ALWAYS AS ( SYSIBM.CLIENT_HOST ) ,
        CHGIPADDR   VARCHAR(128) CCSID 273 GENERATED ALWAYS AS ( SYSIBM.CLIENT_IPADDR ) ,
        CHGPORT     INTEGER                GENERATED ALWAYS AS ( SYSIBM.CLIENT_PORT ) ,
        CHGPCKNAME  VARCHAR(128) CCSID 273 GENERATED ALWAYS AS ( SYSIBM.PACKAGE_NAME ) ,
        CHGPCKSCH   VARCHAR(128) CCSID 273 GENERATED ALWAYS AS ( SYSIBM.PACKAGE_SCHEMA ) ,
        CHGPCKVERS  VARCHAR(64)  CCSID 273 GENERATED ALWAYS AS ( SYSIBM.PACKAGE_VERSION ) ,
        CHGROUTSCH  VARCHAR(128) CCSID 273 GENERATED ALWAYS AS ( SYSIBM.ROUTINE_SCHEMA ) ,
        CHGROUTSPC  VARCHAR(128) CCSID 273 GENERATED ALWAYS AS ( SYSIBM.ROUTINE_SPECIFIC_NAME )
,
        CHGROUTTYP  CHAR(1)      CCSID 273 GENERATED ALWAYS AS ( SYSIBM.ROUTINE_TYPE ) )

        RCDFMT MYADDTABLE ;

Commit;

Insert Into COMSQLQRY.MYADDTABLE (IntVal)
       Values(Midnight_Seconds(Current_Timestamp));

Select * from ComSQLQry.MyAddTable;


-------------------------------------------------------------------------------------
-- Instead Of Trigger
-------------------------------------------------------------------------------------
Drop Table if exists ComSQLQry.Itembas;
Drop Table if exists ComSQLQry.ItemDet;
Commit;

Create Or Replace Table ComSQLQry.ItemBas (
         ItemNo       Integer                Not NULL Default 0,
         Description  Varchar(50) CCSID 1141 Not NULL Default '',
         Status       Char(2)     CCSID 1141 Not NULL Default 'A',
         CrtTimeStmp  Timestamp              Not NULL Default Current_Timestamp,
         UpdTimeStmp  Timestamp              Not NULL Default '0001-01-01-00.00.00.000000')
   Rcdfmt ItemBas;

Create Or Replace Table ComSQLQry.ItemDet (
         ItemNo       Integer                 Not NULL Default 0,
         Color        Char(20)    CCSID 1141  Not NULL Default '',
         Material     Varchar(50) CCSID 1141  Not NULL Default '',
         Height       Decimal(5, 3)           Not NULL Default 0,
         Width        Decimal(5, 3)           Not NULL Default 0,
         Depth        Decimal(5, 3)           Not NULL Default 0,
         "COMMENT"    Varchar(256) CCSID 1141 Not NULL Default '',
         CrtTimeStmp  Timestamp               Not NULL Default Current_Timestamp,
         UpdTimeStmp  Timestamp               Not NULL Default '0001-01-01-00.00.00.000000')
   Rcdfmt ItemDet;

Create Or Replace View ComSQLQry.ItemBV01
       (ItemNo, Description, Status,  Color,   Material,
        Height, Width,      Depth,   Comment, Volume)
    as Select b.ItemNo, Description, Status, Color,   Material,
              Height,   Width,       Depth,  Comment,
              Cast(Height * Width * Depth as Dec(15, 9))
         From ItemBas b Left Outer Join ItemDet d on b.ItemNo = d.ItemNo;

commit;
```

```
Insert into ItemBV01 (ItemNo, Description)
Values(100, 'Rocker Granpa Hugo');

Insert Into ItemBV01 (ItemNo, Description, Color, Comment)
Values(200, 'Couch Aunt Trude', 'Antic Pink', 'Ultra Soft');

Select * from ItemBV01;

Create Or Replace Trigger ComSQLQry.InsertItem
       Instead Of Insert On ComSQLQry.ItemBV01
       Referencing Old As O
                   New As N
      For Each Row
      Mode DB2SQL

      Set Option DbgView=*Source

   Begin Atomic
      Declare Isfound Smallint Default 0;

      Insert Into ComSQLQry.Itembas (Itemno, Description, Status, CrtTimestmp, UpdTimestmp)
            Values( N.Itemno, N.Description, Case When N.Status = ' ' Or N.Status Is Null
                                                  Then 'A'
                                                  Else N.Status
                                             End,
                  Default, Default);

      If    N.Color    Is Not Null
         Or N.Material Is Not Null
         Or N.Height   Is Not Null
         Or N.Width    Is Not Null
         Or N.Depth    Is Not Null
         Or N.Comment  Is Not Null
         Then  Insert Into ComSQLQry.ItemDet
                     (Itemno,    Color, Material, Height, Width, Depth, Comment, CrtTimestmp,
UpdTimestmp)
               Values(N.Itemno,  Coalesce(N.Color,    'Colored'),
                                 Coalesce(N.Material, 'Various'),
                                 Coalesce(N.Height,   0),
                                 Coalesce(N.Width,    0),
                                 Coalesce(N.Depth,    0),
                                 Coalesce(N.Comment,   ''),
                      Default, Default);
      End If;
End;

Commit;
```