

```

///$tab Main
SET ThousandsSep='.';
SET DecimalSep=',';
SET MoneyThousandSep='.';
SET MoneyDecimalSep=',';
SET MoneyFormat='$#.##0,00;-$.##0,00';
SET TimeFormat='h:mm:ss TT';
SET DateFormat='DD/MM/YYYY';
SET TimestampFormat='DD/MM/YYYY h:mm:ss[.fff] TT';
SET MonthNames='ene.;feb.;mar.;abr.;may.;jun.;jul.;ago.;sep.;oct.;nov.;dic.';
SET DayNames='lun.;mar.;mié.;jue.;vie.;sáb.;dom.';

//ODBC CONNECT32 TO ALIANZAS;
////----- Start Multiple Select Statements -----
//LOAD `GIRO_ID`,
//  `GIRO_CANALDEST`,
//  `TRNS_ID`,
//  Replace(`GIRO_VALOR`,`.',',') as GIRO_VALOR,
//  `GIRO_FLETE`,
//  `GIRO_VALORSMS`,
//  `GIRO_ORIGEN`,
//  `GIRO_DESTINO`,
//  `GIRO_REGISTRADOPOR`,
//  `GIRO_FECHAREGISTRO`,
//  `GIRO_CEDULADEST`,
//  `GIRO_REFERENCIA`;
//SQL SELECT *
//FROM ALIANZAS.GIRO
//WHERE GIRO_FECHAREGISTRO BETWEEN '2014-07-01 00:00:00' AND '2014-07-01 23:59:59';
////----- End Multiple Select Statements -----
//
//

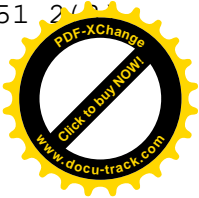
///$tab INICIAL
ODBC CONNECT32 TO GIROS;
/*
modo temporizador puede tener los siguientes valores:
0 Tiempo de ejecución guión
1 Tiempo de llamada de función
2 Hora en que se abre el documento

NOW(0)AS GIRO_FECHAESTADO,
*/

D_Giros_Temp:

LOAD GIRO_ID & 'E-'& ESTADOGIRO_ID as KeyGen,
  GIRO_ID,
  GIRO_LOGIN_ORIGEN,
  GIRO_ORIGEN,
  GIRO_DESTINO,
  GIRO_MONTO,
  GIRO_PAGO,
  GIRO_ENVIO,
  GIRO_DESCRIPCION,
  GIRO_FECHAPAGO,
  GIRO_REGISTRADOPOR,
  GIRO_FECHAENVIO,
  GIRO_FECHAREGISTRO,
  GIRO_FLETE,
  FACTURA_ENVIO,
  FACTURA_PAGO,
  RESPUESTA_ENVIO,
  RESPUESTA_PAGO,
  SENDSMS,
  SMSVALUE,
  IDOPERATOR,
  DIST_ORIGEN,
  DIST_DESTINO,
  ESTADOGIRO_ID,
  CADENA_DESTINO,
  CADENA_ORIGEN,
  GIRO_TPID_REM,
  GIRO_TPIDNOMBRE_REM,
  GIRO_NUMEROIDENTIFICACION_REM,
  GIRO_NOMBRE_REM,
  GIRO_APELLIDO_REM,
  GIRO_DIRECCION_REM,
  GIRO_ZONAIID_REM,

```



```

GIRO_DEPARTAMENTO_REM,
GIRO_CIUADAD_REM,
GIRO_CELULAR_REM,
GIRO_TELEFONO_REM,
GIRO_TPID_DEST,
GIRO_TPIDNOMBRE_DEST,
GIRO_NUMEROIDENTIFICACION_DEST,
GIRO_NOMBRE_DEST,
GIRO_APELLIDO_DEST,
GIRO_DIRECCION_DEST,
GIRO_ZONAIID_DEST,
GIRO_DEPARTAMENTO_DEST,
GIRO_CIUADAD_DEST,
GIRO_CELULAR_DEST,
GIRO_TELEFONO_DEST,
PTO_NOMBRE_ORG,
PTO_DEPARTAMENTO_ORG,
PTO_CIUADAD_ORG,
PTO_DIRECCION_ORG,
PTO_BARRIO_ORG,
PTO_TELEFONO_ORG,
PTO_NOMBRE_DEST,
PTO_DEPARTAMENTO_DEST,
PTO_CIUADAD_DEST,
PTO_DIRECCION_DEST,
PTO_BARRIO_DEST,
PTO_TELEFONO_DEST,
GIRO_LOGIN_DESTINO,
GIRO_REFERENCIA;

```

```

SQL SELECT *
FROM GIROS.GIROS
where GIRO_FECHAREGISTRO between '2014-07-21 00:00:00.0' AND '2014-07-21 23:59:59.0' ;

```

```

STORE D_Giros_Temp INTO ..\Qvd\R_Giros14_Ext.qvd (QVD);
DROP Table D_Giros_Temp;

```

```

///$tab Incretetal Giros

```

```

SET vQvdFile='..\Qvd\R_Giros14_Ext.qvd'; ///? El Nombre del archivo QVD
SET vTableName='D_Giros_Temp'; ///? Nombre de mi tabla
SET vPK='GIRO_ID'; ///? Clave principal de datos

```

```

LET vQvdExists = if(FileSize('${vQvdFile}') > 0, -1, 0);
// Establecer una variable que indica si el QVD existe o no. -1 Es verdadero, 0 es False
IF ${vQvdExists} THEN // QVD existe, vamos a hacer un reload incrementales
maxdateTab:
// Obtener la fecha máxima de esta QVD para su uso en SELECT incrementales
LOAD max(Timestamp(GIRO_FECHAREGISTRO,'YYYY-MM-DD h:mm:ss')) as maxdate ///? Establecer el nombre del campo
FROM ${vQvdFile} (qvd);
///? Fecha puede necesitar ser formateada para que coincida con el formato esperado por el SELECT de S
///? por ejemplo, la fecha (peek ('MaxDate'), 'MMDDAAAA');
LET vIncrementalExpression = peek('maxdate');
DROP table maxdateTab;

```

```

ELSE // QVD no existe
LET vIncrementalExpression = 'WHERE ModDate >=' & peek('maxdate'); // No QVD. Force full reload
// LET vIncrementalExpression = 'DONDE ModDate> =' & MAKEDATE (2005,01,01); / ///? También se pudo esta
END IF

```

```

///$tab Data Load

```

```

ODBC CONNECT32 TO GIROS;

```

```

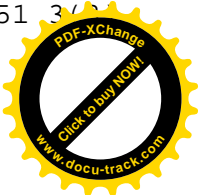
${vTableName}:

```

```

LOAD GIRO_ID & 'E-' & ESTADOGIRO_ID as KeyGen,
GIRO_ID,
GIRO_LOGIN_ORIGEN,
GIRO_ORIGEN,
GIRO_DESTINO,
GIRO_MONTO,
GIRO_PAGO,
GIRO_ENVIO,
GIRO_DESCRIPCION,
GIRO_FECHAPAGO,
GIRO_REGISTRADOPOR,

```



```

GIRO_FECHAENVIO,
GIRO_FECHAREGISTRO,
GIRO_FLETE,
FACTURA_ENVIO,
FACTURA_PAGO,
RESPUESTA_ENVIO,
RESPUESTA_PAGO,
SENDSMS,
SMSVALUE,
IDOPERATOR,
DIST_ORIGEN,
DIST_DESTINO,
ESTADOGIRO_ID,
CADENA_DESTINO,
CADENA_ORIGEN,
GIRO_TPID_REM,
GIRO_TPIDNOMBRE_REM,
GIRO_NUMEROIDENTIFICACION_REM,
GIRO_NOMBRE_REM,
GIRO_APELLIDO_REM,
GIRO_DIRECCION_REM,
GIRO_ZONAIID_REM,
GIRO_DEPARTAMENTO_REM,
GIRO_CIUADAD_REM,
GIRO_CELULAR_REM,
GIRO_TELEFONO_REM,
GIRO_TPID_DEST,
GIRO_TPIDNOMBRE_DEST,
GIRO_NUMEROIDENTIFICACION_DEST,
GIRO_NOMBRE_DEST,
GIRO_APELLIDO_DEST,
GIRO_DIRECCION_DEST,
GIRO_ZONAIID_DEST,
GIRO_DEPARTAMENTO_DEST,
GIRO_CIUADAD_DEST,
GIRO_CELULAR_DEST,
GIRO_TELEFONO_DEST,
PTO_NOMBRE_ORG,
PTO_DEPARTAMENTO_ORG,
PTO_CIUADAD_ORG,
PTO_DIRECCION_ORG,
PTO_BARRIO_ORG,
PTO_TELEFONO_ORG,
PTO_NOMBRE_DEST,
PTO_DEPARTAMENTO_DEST,
PTO_CIUADAD_DEST,
PTO_DIRECCION_DEST,
PTO_BARRIO_DEST,
PTO_TELEFONO_DEST,
GIRO_LOGIN_DESTINO,
GIRO_REFERENCIA;

```

```

SQL SELECT *
FROM GIROS.GIROS
WHERE GIRO_FECHAREGISTRO >= '${vIncrementalExpression}'
;
///$tab Incremental Update

```

#### Directory;

```

// If incremental reload was , load previous data and concatenate to data just read.
// Si recarga incremental fue de, cargar los datos anteriores y concatenar los datos sólo leer.
IF ${vQvdExists} THEN
    // Use CONCATENATE in case we've added any new fields.
    // Usar concatenar en caso hemos añadido nuevos campos nuevos.
    CONCATENATE (${vTableName}) LOAD * FROM ${vQvdFile} (qvd)
    WHERE NOT exists(${vPK}) // Load only QVD rows that were not already loaded in the data load.
    // Cargar sólo las filas QVD que no estaban ya cargados en la carga de datos.
;
END IF

```

```

STORE ${vTableName} INTO ${vQvdFile};
DROP table ${vTableName} ;

```

```

///$tab Actualizar Data
///$tab EXIT
EXIT Script

```