

Base de Datos: Repositorio de Carga de Script: Plantillas Contables

Esta sección está dedicada a la carga de plantillas contables.

Contexto de la solución

La carga se realizará en los siguientes pasos:

- Crear Tabla temporal
- Cargar archivo
- Validaciones previas
- Ejecutar proceso
- Validaciones posteriores

Crear Tabla temporal

```
CREATE TABLE CONTABIL01.TMP_PLANTILLAS
(
  *CODIGO                                VARCHAR2(32 BYTE),
  *DESCRIPCION                           VARCHAR2(64 BYTE),
  *TIPO_CCOSTO                            VARCHAR2(128 BYTE),
  *CUENTA_DEBITO                          VARCHAR2(32 BYTE),
  *CUENTA_CREDITO                         VARCHAR2(32 BYTE),
  CUENTA_IVA                             VARCHAR2(32 BYTE),
  PORCENTAJE_IVA                          NUMBER(5,2),
  TERCERO                                NUMBER(10),
  INACTIVA                                CHAR(1 BYTE) DEFAULT 'N',
  BANCO_CTA_DEBITO                        VARCHAR2(32 BYTE),
  BANCO_CTA_CREDITO                       VARCHAR2(32 BYTE),
  CUENTA_ORDEN_DEBITO                     VARCHAR2(32 BYTE),
  CUENTA_ORDEN_CREDITO                    VARCHAR2(32 BYTE),
  CODIGO_CUENTA_AMORTIZA                   VARCHAR2(32 BYTE),
  NATURALEZA_TERCERO                       CHAR(1 BYTE) DEFAULT 'A'
  NOT NULL,
  USUARIO_EMPRESA                         VARCHAR2(30 BYTE) DEFAULT USER,
  CODIGO_MEMPRESA                         VARCHAR2(50 BYTE) DEFAULT '999999999',
  CODIGO_PLANTILLA_HIJA                     NUMBER,
  CUENTA_DIF_DEBITO                        VARCHAR2(32 BYTE),
  CUENTA_DIF_CREDITO                       VARCHAR2(32 BYTE),
  DISTRIBUIR_PORCENTAJES_DEB_CRE           VARCHAR2(1 BYTE) DEFAULT 'S',
  TIPO_CAUSACION                           NUMBER,
  TIPO_PLANTILLA                           NUMBER
);
```

--* Columnas cubiertas en la carga en esta versión liberada

Cargar archivo

En este paso se utilizan herramientas como golden el cual permite carga la tbla con un simple copiar y pegar. Si no se tiene a la mano la herramienta se debe convertir el archivo de carga a .csv e importar los datos en la tabla desde un ide como Toad, SQLDeveloper, DBeaver, etc...

Validaciones previas

Si las siguiente consultas devuelven al menos 1 registros. El proceso no se puede continuar y debe evaluar la data reportada por parte del consultor o cliente.

```
SELECT  t.codigo, T.CUENTA_DEBITO
FROM    CONTABIL01.TMP_PLANTILLAS t
WHERE   t.codigo NOT IN (SELECT codigo FROM contabil01.plantillas)
AND NOT EXISTS (SELECT 1 FROM TESORE01.PLAN_CUENTAS_CONTABLES pcc WHERE
PCC.CODIGO_CUENTA = T.CUENTA_DEBITO )
ORDER BY 1 ASC;

SELECT  t.codigo, T.CUENTA_CREDITO
FROM    CONTABIL01.TMP_PLANTILLAS t
WHERE   t.codigo NOT IN (SELECT codigo FROM contabil01.plantillas)
AND NOT EXISTS (SELECT 1 FROM TESORE01.PLAN_CUENTAS_CONTABLES pcc WHERE
PCC.CODIGO_CUENTA = T.CUENTA_CREDITO )
ORDER BY 1 ASC;

SELECT  t.codigo, T.TIPO_CCOSTO
FROM    CONTABIL01.TMP_PLANTILLAS t
WHERE   t.codigo NOT IN (SELECT codigo FROM contabil01.plantillas)
AND NOT EXISTS (SELECT 1 FROM NOMINA.TIPOS_NOMINA tn WHERE
TN.S_DESCRIPCION = T.TIPO_CCOSTO)
ORDER BY 1 ASC;
```

Ejecutar proceso

```
DECLARE
--
=====
==
-- Fecha: 2:22 p. m. lunes, 11 de julio de 2022 - carlos.torres@ada.co
-- Extrae los códigos de la carga y la cantidad de coincidencias
--
=====
```

```

==
Cursor curCodigos
IS
SELECT  t.codigo, t.descripcion, COUNT(t.codigo)
FROM    CONTABIL01.TMP_PLANTILLAS t
WHERE   t.codigo NOT IN (SELECT codigo FROM contabil01.plantillas)
GROUP BY t.codigo, t.descripcion
ORDER BY 1 ASC;
--

=====

==
-- Fecha: 2:22 p. m. lunes, 11 de julio de 2022 - carlos.torres@ada.co
-- Consulta el contenido de cada código
--

=====

==
Cursor tmpPlantillas(as_codigo varchar2)
IS
SELECT  t.*
FROM    CONTABIL01.TMP_PLANTILLAS t
WHERE   t.codigo = as_codigo;

--

=====

==
-- Fecha: 2:24 p. m. lunes, 11 de julio de 2022 - carlos.torres@ada.co
-- Declaraciones Locales
--

=====

==
lvn_tipo_ccostos NUMBER;
lvn_cuenta_debito NUMBER;
lvn_cuenta_credito NUMBER;
lvn_seq_plantilla NUMBER;
lvn_seq_det_plantilla NUMBER;
lva_mensaje varchar2(1024);
BEGIN
--

=====

==
-- Fecha: 9:37 a. m. jueves, 7 de julio de 2022 - carlos.torres@ada.co
-- Recorrer el cursor de registros temporales
--

=====

==
lva_mensaje := PCK_UTILIDADES.F_MSG('Recorrer el cursor de registros
temporales');
FOR lregCodigo IN curCodigos
Loop
--

=====

```

```
==
-- Fecha: 2:26 p. m. lunes, 11 de julio de 2022 - carlos.torres@ada.co
-- Generar el identificador de la plantilla
--

=====

==
lva_mensaje := PCK_UTILIDADES.F_MSG('Generar el identificador para la
plantilla código: ' || lregCodigo.codigo);
SELECT  CONTABIL01.SEQ_PLANTILLAS.nextval
INTO    lvn_seq_plantilla
FROM    DUAL;
--

=====

==
-- Fecha: 9:40 a. m. jueves, 7 de julio de 2022 - carlos.torres@ada.co
-- Insertar Encabezado de la plantilla
--

=====

==
lva_mensaje := PCK_UTILIDADES.F_MSG('Insertar Encabezado de la plantilla
' || lregCodigo.codigo);
INSERT INTO CONTABIL01.PLANTILLAS (CODIGO_INTERNO, CODIGO, DESCRIPCION)
VALUES (lvn_seq_plantilla, lregCodigo.codigo, lregCodigo.descripcion);
FOR lregTmp IN tmpPlantillas(lregCodigo.codigo)
Loop
--

=====

==
-- Fecha: 9:41 a. m. jueves, 7 de julio de 2022 - carlos.torres@ada.co
-- Identificar el tipo de centro de costos
--

=====

==
lva_mensaje := PCK_UTILIDADES.F_MSG('Identificar el código interno del
tipo de centro de costos: ' || lregTmp.tipo_ccosto);
SELECT  KA_NL_TIPO_NOMINA
INTO    lvn_tipo_ccostos
FROM    NOMINA.TIPOS_NOMINA
WHERE   S_DESCRIPCION = lregTmp.tipo_ccosto;
--

=====

==
-- Fecha: 9:46 a. m. jueves, 7 de julio de 2022 - carlos.torres@ada.co
-- Identificar cuenta debito
--

=====

==
lva_mensaje := PCK_UTILIDADES.F_MSG('Identificar el código interno de
la cuenta debito: ' || lregTmp.CUENTA_DEBITO);
SELECT  PCC.CODIGO_INTERNO_CUENTA
```

```

INTO    lvn_cuenta_debito
FROM    TESORE01.PLAN_CUENTAS_CONTABLES PCC
WHERE   PCC.CODIGO_CUENTA = lregTmp.CUENTA_DEBITO
AND     PCC.TIPO_PLAN_CONTABLE = 1;
--

=====

==
-- Fecha: 9:50 a. m. jueves, 7 de julio de 2022 - carlos.torres@ada.co
-- Identificar cuenta crédito
--

=====

==
    lva_mensaje := PCK_UTILIDADES.F_MSG('Identificar el código interno de
la cuenta crédito: ' || lregTmp.CUENTA_CREDITO);
SELECT  PCC.CODIGO_INTERNO_CUENTA
INTO    lvn_cuenta_credito
FROM    TESORE01.PLAN_CUENTAS_CONTABLES PCC
WHERE   PCC.CODIGO_CUENTA = lregTmp.CUENTA_CREDITO
AND     PCC.TIPO_PLAN_CONTABLE = 1;
--

=====

==
-- Fecha: 9:51 a. m. jueves, 7 de julio de 2022 - carlos.torres@ada.co
-- Insertar el detalle de la plantilla contable
--

=====

==
    lva_mensaje := PCK_UTILIDADES.F_MSG('Insertar el detalle de la
plantilla contable: ' || lregCodigo.codigo || ' y tipo de centro de costos:
' || lregTmp.tipo_ccosto);
SELECT  CONTABIL01.SEQ_DET_CUENTA_PUC.nextval
INTO    lvn_seq_det_plantilla
FROM    DUAL;
--

=====

==
-- Fecha: 10:03 a. m. jueves, 7 de julio de 2022 -
carlos.torres@ada.co
-- Insertar detalle de cuentas contables
--

=====

==
    lva_mensaje := PCK_UTILIDADES.F_MSG('Insertar detalle de cuentas
contables de la plantilla contable: ' || lregCodigo.codigo || ' y tipo de
centro de costos: ' || lregTmp.tipo_ccosto);
INSERT INTO CONTABIL01.DET_PLANTILLAS_CUENTAS_PUC (CODIGO_PLANTILLA,
CODIGO, TIPO_PLAN_CONTABLE, TIPO_CCOSTO, CUENTA_DEBITO, CUENTA_CREDITO)
VALUES (lvn_seq_plantilla, lvn_seq_det_plantilla, 1, lvn_tipo_ccostos,
lvn_cuenta_debito, lvn_cuenta_credito);
Commit;
END Loop;

```

```
END Loop;
exception WHEN others THEN
    PCK_UTILIDADES.P_REGISTER_ERROR(SQLCODE, SQLERRM, lva_mensaje || ' - ' ||
    DBMS_UTILITY.FORMAT_ERROR_BACKTRACE, 'BLOQUE ANONIMO');
    ROLLBACK;
END;
```

Validaciones posteriores

Se definen consultas que evidencian si el proceso de carga se realizó correctamente. En este caso hay una consulta que indica cuantos se cargaron y cuantos no.

```
SELECT 'No Cargada' AS label, t.codigo, t.descripcion, COUNT(t.codigo)
FROM    CONTABIL01.TMP_PLANTILLAS t
WHERE   t.codigo NOT IN (SELECT codigo FROM contabil01.plantillas)
GROUP BY t.codigo, t.descripcion
UNION
SELECT 'Cargada' AS label, t.codigo, t.descripcion, COUNT(t.codigo)
FROM    CONTABIL01.TMP_PLANTILLAS t
WHERE   t.codigo IN (SELECT codigo FROM contabil01.plantillas)
GROUP BY t.codigo, t.descripcion
ORDER BY 1,2,3;
```

Notas del Proceso

- Sólo se soportan las columnas con asterisco en esta versión.
- El proceso es autocommit.
- Se puede relanzar y solo aplicará para los registros faltantes.

[←Volver atras](#)

From:
<http://wiki.adacsc.co/> - Wiki

Permanent link:
<http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:database:repositorioscriptsload:plantillascontables>

Last update: 2022/07/11 20:25