



LOGALI

TEORÍA

ABAP OPEN SQL - Base de datos

SAP ABAP Programación





Contenido

1. Definiciones	3
2. Elementos comunes a la creación de tablas transparentes y estructuras	3
2.1. Dominio.....	3
2.2. Elemento de datos.....	5
2.3. Tabla transparente	6
2.4. Estructura.....	10
3. Añadir registros en una tabla transparente	11
4. Visualizar la información almacenada en una tabla transparente	12
5. Los datos en el sistema SAP	13
5.1. Datos maestros:.....	13
5.2. Datos de movimientos:	13
5.3. Datos del sistema o de control:.....	13
6. Instrucciones SQL de ABAP/4.....	13
6.1. SELECT	14
6.2. INSERT.....	14
6.3. UPDATE.....	15
6.4. MODIFY	16
6.5. DELETE.....	17



1. Definiciones

Estructuras: es un conjunto de definiciones de campos almacenado en el diccionario de SAP. Una estructura nunca podrá “llenarse” con ningún registro de información asociada en la base de datos. Su uso se sitúa solamente dentro de la programación ABAP.

Tabla transparente: al igual que una estructura, una tabla transparente también corresponde a un conjunto definido de campos estructurados almacenada en el diccionario de SAP. La diferencia es que una tabla transparente almacena información. De hecho, una tabla en general es la unidad mínima necesario para introducir información en nuestra base de datos.

2. Elementos comunes a la creación de tablas transparentes y estructuras

Previo a crear una tabla o estructura hay que definir los campos que contendrán éstas. Cada campo de una tabla o estructura ha de tener una descripción y una definición del tipo de información que contendrá. En SAP esto se consigue mediante dos objetos del diccionario llamados *dominio* y *elemento de datos*.

2.1. Dominio

El dominio se define como el rango de valores que puede tener un campo. Es decir, el tamaño máximo que puede tener un campo (desde 0 caracteres o números hasta lo que se nos ocurra) y el tipo de información que podrá albergar (caracteres, números enteros, números decimales, etc.). Por ejemplo, creemos aquí el dominio para el campo ‘código’.

Herramientas / Workbench ABAP / Desarrollo / SE11 – Dictionary ABAP / Dominio



Marcamos la opción 'dominio', lo llamaremos 'ZD_CODIGO_USUARIO'. Pulsamos 'Crear'. (Reemplazar USUARIO con el nombre de su usuario SAP para evitar conflictos de nomenclatura con otros usuarios)

Damos una descripción al dominio.

Escogemos un tipo de datos. En este caso sólo aceptará datos numéricos así que voy a marcar la opción 'NUMC', que sólo acepta números enteros. Así evitaremos que el usuario se pueda equivocar e introduzca letras en el campo. SAP validará que el tipo de información es numérica y no dejará grabar si no lo es.

Aceptaremos hasta 9999 contactos. Por lo tanto, la cantidad de posiciones será de 4. La longitud de salida también será de 4. Es decir, lo que veremos en la tabla será lo mismo que entremos en el registro.

En la pestaña 'Ámbito val' podemos definir los valores fijos que puede tomar el campo. En este caso, lo dejamos vacío. Pero en el caso de que el campo sólo pudiese tomar dos valores como 'verdadero' o 'falso', entonces lo podríamos indicar aquí y SAP al entrar la información luego en la tabla no nos dejaría entrar ningún otro valor que no fuese verdadero o falso. No es nuestro caso. Grabamos y asignamos la orden de transporte. Activamos.

Dictionary: Modificar dominio

Dominio: ZD_CODIGO_LOGALI nuevo(revisado)

Descripción breve: Código

Atributos Definición Ámbito val.

Formato

Tipo de datos: NUMC String sólo con cifras

Ctd.posiciones: 8

Decimales: 0

Propiedades salida

Longitud salida: 8

Rutina conv.

☐ Signo +/-

☐ Minúsculas



2.2. Elemento de datos

El elemento de datos contiene la suma del dominio y la descripción del campo.

Herramientas / Workbench ABAP / Desarrollo / SE11 – Dictionary ABAP / Tipo datos / Elem. datos

Marcamos la opción ‘tipo de datos’. Escribimos el código del elemento de datos. Pulsamos ‘crear’. Nos muestra un desplegable con tres opciones. Seleccionamos la opción ‘elemento de datos’.

De nuevo damos una descripción y escribimos el dominio que hemos creado anteriormente. Otra opción es utilizar el botón ‘tipo instalado’. Con esta opción no utilizaríamos ningún elemento de datos, sino que asignamos el tipo de datos y su longitud directamente en el elemento de datos. No es nuestro caso. En la pestaña de ‘denominación de campo’ escribimos cuatro descripciones adicionales. Serán las que aparecerán en el encabezado de la tabla según el tamaño del campo. Guardamos y activamos.

Dictionary: Modificar elemento datos

Elemento datos: ZELEMENTO_LOGALI nuevo(revisado)

Descripción breve: Código de contacto

Tablas: Atributos Tipo datos Propiedades adicionales Denom. campo

	Long.	Denominador de campo
Breve	<input type="checkbox"/>	Código
Mediano	<input type="checkbox"/>	Código de contacto
Largo	<input type="checkbox"/>	Código de contacto
Cabecera	<input type="checkbox"/>	Código de contacto



2.3. Tabla transparente

Una vez creados dominios y elementos de datos podemos empezar a crear nuestra tabla que servirá de contenedor de información dentro de la base de datos SAP.

Herramientas / Workbench ABAP / Desarrollo / SE11 – Dictionary ABAP / Tabla base de datos

Al presionar el botón ‘Crear’ nos pedirá una descripción y un tipo de clase de entrega. Nuestra tabla almacenará información así que la definimos como ‘A’ (tabla de aplicación). También permitiremos que se puedan añadir registros directamente así que marcaremos la opción ‘X Visualización/actualización permitida’.



Dictionary: Modificar tabla

Tabla transparente: ZCONTACT_LOGALI nuevo(revisado)

Descripción breve: Tabla contactos

Atributos Entrega y actualización Campos Ayuda p./Verif.entr. Campos de moneda/cantidad

Clase de entrega: A Tabla aplicación (datos maestros y de movimiento)

Browser datos/Actual.vista tabla: Visual./Actual.permitida

Añadimos los nombres de cada campo. A cada campo asignamos su elemento de datos si lo tenemos. En el caso del código tenemos el elemento creado y se lo asignamos. En el caso del nombre no lo tenemos, pero vamos a utilizar alguno de los que existen en SAP de forma estándar. Buscamos a través del matchcode un elemento de datos estándar de tipo carácter y tamaño 40.

Tabla Tratar Pasar a Utilidades Detalles Entorno Sistema Ayuda

Dictionary: Modificar tabla

Tabla transparente: ZCONTACT_LOGALI nuevo(revisado)

Descripción breve: Tabla contactos

Atributos Entrega y actualización Campos Ayuda p./Verif.entr. Campos de moneda/cantidad

Campo	Clv	Val...	Elem.datos	Tipo de d...	Long.	Deci...	Descripción breve
MANDT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	MANDT	CLNT	3	0	Mandante
NOMBRE	<input checked="" type="checkbox"/>	<input type="checkbox"/>	TEXT	CHAR	29	0	Campo de texto
DIRECCION	<input type="checkbox"/>	<input type="checkbox"/>	TEXT	CHAR	29	0	Campo de texto
	<input type="checkbox"/>	<input type="checkbox"/>					
	<input type="checkbox"/>	<input type="checkbox"/>					
	<input type="checkbox"/>	<input type="checkbox"/>					
	<input type="checkbox"/>	<input type="checkbox"/>					

En el caso de dirección y teléfono no utilizaremos ningún elemento de datos. Asignaremos el tipo de datos, longitud y descripción directamente en la tabla. Para eso pulsamos el botón 'Tipo instalados' y escribimos la información directamente.



Determino cuáles son los campos clave, es decir, aquellos que determinarán un registro como único en la tabla. En esta tabla esos campos son el campo 'mandante' y el campo 'código'. Esto significa que el código de cada contacto es unívoco y no pueden aparecer dos registros con el mismo código dentro de la base de datos. El campo mandante es necesario siempre en todas las tablas transparentes para indicar que la información contenida en una tabla transparente será diferente según el mandante en el cual nos encontremos. Para eso marco la columna 'Clv' de estos campos.

Actualizamos las opciones técnicas: en *Pasar a / Opciones técnicas* indicamos la clase de datos que informaremos a nuestra tabla. En el ejemplo, los datos que incluirá serán información de datos maestros (APPL0). El tamaño será el mínimo posible (0 – de 0 a 3300), seguro que no incluiré más registros. El resto de información de esta pantalla la mantengo tal cual.



Dictionary ABAP: Actualizar opciones técnicas

Revisado<-> Activo

Nom. ZCONTACT_LOGALI Transparente Tabla

Descrip.breve Tabla contactos

Últ.modificación LOGALI 07.10.2015

Status nvo no grabados

Propiedades generales Propiedades específicas BD

Parámetros de memoria

Clase de datos

Categ.tamaño

Grabación en memoria int

☒ Grabación en memoria

☐ Permitido grabar en MI

☐ Grabación en memoria

Forma de grabación en memoria intermedia

☐ Reg.ind.grabados en MI

☐ Área genérica grabada en MI Cantidad campos clave

☐ Grabado completamente en MI

☐ Grabar en log modificaciones de datos

☐ Acc.escritura solo con Java

Entradas posibles: Clase de datos

Cl.datos	Descripción
APPL0	Datos maestros, tablas transparentes
APPL1	Datos variables, tablas transparentes
APPL2	Organización y Customizing
DDIM	Tablas de dimensión en BW
DFACT	Tablas de hechos en BW
DODS	Tablas ODS en BW

☒ Clases de datos de sistema

Actualizamos la categoría de ampliación: si en el futuro queremos ampliar la estructura de la tabla debemos decidir cómo la vamos a ampliar. En el ejemplo, seleccionamos 'ampliable de cualquier manera'. La propia ayuda de la opción nos puede guiar sobre cómo hacer esto. Menú *Detalles / Categoría de ampliación*.



Dictionary ABAP: Actualizar opciones técnicas

Revisado<->Activo

Nom.	ZCONTACT_LOGALI	Transparente Tabla
Descrip.breve	Tabla contactos	
Últ.modificación	LOGALI	07.10.2015
Status	nvo	grabados

Propiedades generales Propiedades específicas BD

Parámetros de memoria lógicos

Clase de datos: APPL0 Datos maestros, tablas transparentes

Categ.tamaño: 0

Grabación en memoria intermedia

☒ Grabación en memoria intermedia no permitida
☐ Permitido grabar en MI, pero desactivado
☐ Grabación en memoria intermedia activada

Forma de grabación en memoria intermedia

☐ Reg.ind.grabados en MI
☐ Área genérica grabada en MI Cantidad campos clave:
☐ Grabado completamente en MI

☐ Grabar en log modificaciones de datos
☐ Acc.escritura solo con Java

Grabamos y activamos.

2.4. Estructura

Básicamente lo que debemos hacer es exactamente lo mismo que hacemos al crear una tabla transparente. Sin embargo, el camino del menú será diferente.

Herramientas / Workbench ABAP / Desarrollo / SE11 – Dictionary ABAP / Tipo datos / Estructura



Directamente debemos añadir los campos con su tipología (columna tipo componente) sin asignar ningún tipo de clase de entrega. No es necesario añadir el campo 'mandante'.

3. Añadir registros en una tabla transparente

Para dotar de contenido una tabla podemos utilizar diversas alternativas directas si hemos marcado la opción de 'browser de datos/actual.vista de tabla' de la pestaña 'Entrega y actualización' como 'X'.

➤ A través del explorador de datos de tablas.

Herramientas / Workbench ABAP / Resumen / SE16 – Browser de datos / Tabla / Registrar entradas

The screenshot shows the SAP SE16 transaction 'Tabla ZCONTACT_LOGALI insertar'. The form has a menu bar with 'Entrada tabla', 'Tratar', 'Pasar a', 'Opciones', 'Entorno', 'Sistema', and 'Ayuda'. Below the menu is a toolbar with various icons. The main area contains a 'Reinicializar' button and four input fields: 'MANDT' with the value '811', 'NOMBRE' with the placeholder 'Nombre Apellidos', 'DIRECCION' with the placeholder 'CALLE / N°', and 'TELEFONO' with the value '912345678'. The 'TELEFONO' field is highlighted with a red box.

➤ A través del browser de visualización general de tabla: mediante la transacción SE16N

- Botón ejecutar.
- Botón añadir líneas.
- Botón grabar.

➤ Creando una vista de tablas y utilizar esta vista para añadir registros.
Herramientas / Workbench ABAP / Desarrollo / SE11 – Dictionary ABAP / Utilidades / Generador actualiz.tablas



También podemos utilizar código ABAP. Utilizando sentencias de SQL como *UPDATE*, *MODIFY* o *INSERT* dentro de nuestro programa podemos ir añadiendo registros a la tabla que hayamos creado. En este caso no es necesario haber marcado la opción de 'browser de datos/actual.vista de tabla' de la pestaña 'Entrega y actualización' como 'X'.

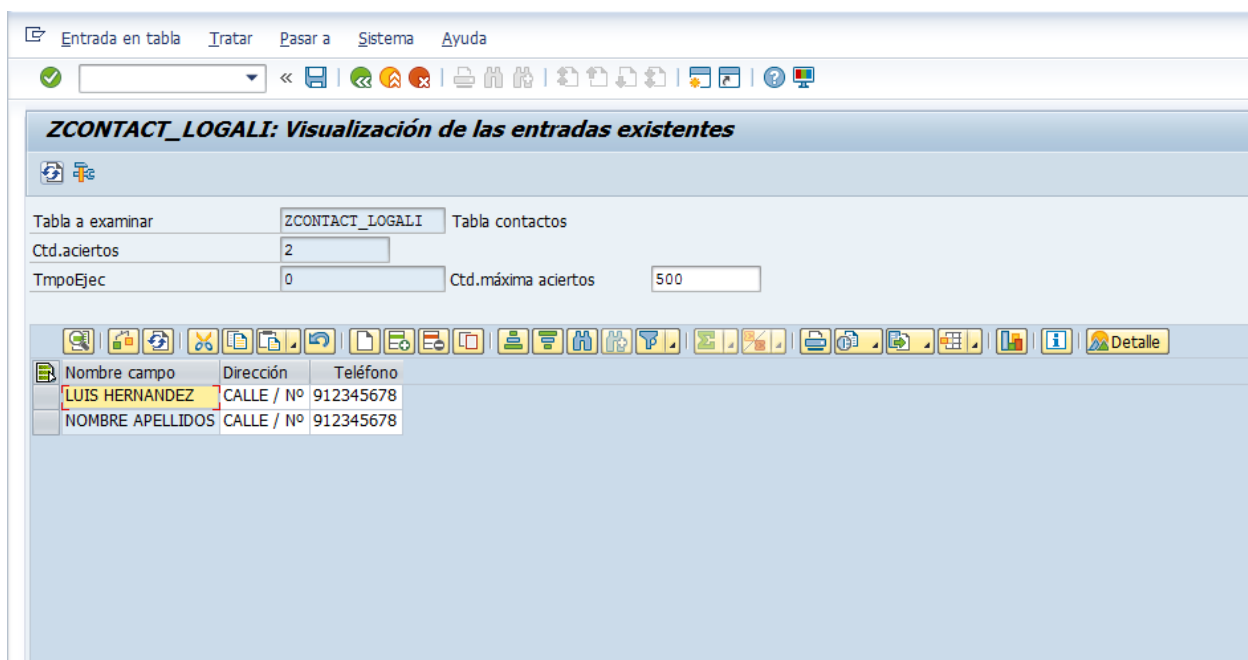
4. Visualizar la información almacenada en una tabla transparente

De nuevo hay dos opciones:

- A través del explorador de datos de tablas.

Herramientas / Workbench ABAP / Resumen / SE16 – Browser de datos

- A través del nuevo browser de visualización en la transacción SE16N.





5. Los datos en el sistema SAP

Podemos clasificar los datos del sistema en datos maestros, datos de movimientos, y datos del sistema.

5.1. Datos maestros:

Son datos que no se modifican muy a menudo. Ej: Materiales, Cuentas, Bancos, Clientes, etc. Se almacenan en tablas transparentes

5.2. Datos de movimientos:

Datos muy volátiles y con gran volumen de generación. Ej: Facturas, Pedidos...

Se suelen guardar en tablas tipo CLUSTER todos ellos con formato parecido (documentos).

5.3. Datos del sistema o de control:

Muchas tablas con pocos datos. Se suelen guardar en tablas de tipo POOL.

6. Instrucciones SQL de ABAP/4.

ABAP/4 tiene un subconjunto de sentencias SQL para su aplicación sobre tablas de la base de datos SAP. estas son:

SELECT, INSERT, UPDATE, MODIFY, DELETE, COMMIT WORK, ROLLBACK WORK.

que trabajan con de las variables del sistema:

SY -SUBRC: Código de retorno de una operación.



SN-DBCNT: Cantidad de registros afectados por la operación procesada.

6.1. SELECT

La sentencia SELECT será la instrucción fundamental para leer información de la base de datos.

SELECT * FROM <tab> INTO TABI,E <intab> (WHERE <cond>)

Llena la tabla interna <intab> machacando los registros que pudiera tener ésta. Si queremos que respete los registros que tenía la tabla **interna** antes de realizar el SELECT tendremos que utilizar:

**SELECT * FROM <tab> APPENDING TABLE <intab>
(WHERE <cond>).**

Podemos indicar un orden en el proceso de selección de registros.

SELECT * ... ORDER BY <campol> <campo2> ...

Si queremos seleccionar un registro para bloquearlo de posibles modificaciones.

SELECT SINGLE FOR UPDATE * FROM <tab>.

6.2. INSERT.

La sentencia **INSERT** permite introducir registros sencillos o el contenido de una tabla interna en una base de datos SAP.

INSERT <tab>.



Grabará en la BDD el registro de cabecera. Por tanto, previamente a esta instrucción moveremos los valores que queremos introducir sobre el área de trabajo de la tabla.

Si **SY-SUBRC** = 0 Registro insertado.

Si **SY-SUBRC** > 0 La clave del registro que queríamos insertar ya existía en la tabla.

También es posible introducir datos desde una tabla interna.

INSERT <tab> FROM TABLE <intab>

Si **SY-SUBRC** = 0 Registros insertados.

Si existe algún registro en la base de datos con clave igual a algún registro de la tabla interna, se producirá un error de ejecución del programa.

La tabla interna podrá tener la misma estructura que la tabla de base de datos utilizando **INCLUDE STRUCTURE** en su declaración.

6.3. UPDATE.

La sentencia **UPDATE** permite modificar el contenido de uno o varios registros.

UPDATE <tab>.

Modifica el registro de la base de datos que está especificado en el registro de cabecera.

Si queremos modificar el contenido de más de un registro a la vez:



UPDATE <tab> SET <campo> = <valor> WHERE <cond>.

Con este UPDATE, todos los registros que cumplan <cond> modificarán el contenido del <campo> por <valor>.

También es posible utilizar la cláusula SET con

<campo> = <campo> + <valor>

o

<campo> = <campo> - <valor>

Es posible modificar registros desde una tabla interna:

UPDATE <tab> FROM TABLE <intab>.

Si el sistema no puede actualizar un registro, el proceso no finalizara sino que continuará con el siguiente registro.

Si **SY-SUBRC = 0** Todos los registros modificados.

Si **SY-SUBRC = 4** No todos los registros han sido modificados.

En **SY-DBCNT** Tendremos la cantidad de registros modifacados.

6.4. MODIFY

La sentencia **MODIFY** se utilizará cuando no estemos seguros si utilizar un INSERT o un UPDATE. Es decir, cuando no sepamos con certeza si un registro existe o no, para modifcarlo o añadirlo.

MODIFY <tab>.

MODIFY<tab> FROM TABLE <intab>



En caso de que sepamos si existe o no un registro, por eficacia utilizaremos INSERTs o UPDATEs.

6.5. DELETE

Para realizar borrados de datos se aplica la sentencia **DELETE**.

DELETE <tab>.

Borrará el registro que especifiquemos en el área de trabajo.

Para borrar más de un registro (todos los que cumplan una cierta condición).

DELETE FROM<tab> WHERE <cond>.

Podemos borrar de BDD todos los registros de una tabla interna.

DELETE FROM <tab> FROM TABLE <intab>.

Si **SY-SUBRC = 0** Todos los registros han sido borrados.

Si **SY-SUBRC = 4** No todos los registros han sido borrados.

En **SY-DBCNT** Tendremos la cantidad de registros borrados.