

Matheus Vitoreli de Oliveira

Unidad 5

Ejercicio 1

Tarea1

Open SQL es un subconjunto y una variación de ANSI SQL. El intérprete ABAP / 4 pasa todas las sentencias Open SQL a la parte de interfaz de base de datos del proceso de trabajo. Allí, ellos se convierten en SQL nativo de la RDMS instalado. Por ejemplo, si se ejecuta una base de datos Oracle, su código Open SQL de ABAP / 4 sería convertido por la interfaz de base de datos a las declaraciones de Oracle SQL, si utiliza Open SQL, las sentencias SQL se pasarán a la interfaz de base de datos el uso de Open SQL tiene tres ventajas principales.

Ventaja: Por lo tanto, las ventajas son qué en las sentencias Open SQL se transforman en sentencias SQL acordes a la base de datos utilizada, esto es gracias a la «Interface de Base de Datos»; está integrada en el «ABAP runtime enviroment», el cual incluye el hardware, sistema operativo y la plataforma independiente de la base de datos, la interface de base de datos es la encargada de ser el puente entre el servidor de aplicación y el servidor de base de datos.

Implementar: SQL abierto desde la interfaz de la base de datos y, a continuación, se pasa a la base de datos. Cuando se transforman las sentencias, se insertan en el código partes específicas de ABAP, como las condiciones para la gestión automática de clientes o el control de acceso CDS.

Disponible en: < <https://abapmachine.com/abap-open-sql-sentencias-sql/>>.
Acesso em: 27 mar. 2023.

Ejercicio 2

Tarea 1

ABAP Editor: Change Report ZAPPATEST8

Report: ZAPPATEST8 Active

```
1  *~-----*
2  *~ Report  ZAPPATEST8
3  *~
4  *~-----*
5  *~
6  *~
7  *~-----*
8
9  REPORT  ZAPPATEST8.
10 tables spfli.
11 select * from spfli into spfli order by carrid.
12       write: spfli-cityfrom, spfli-cityto.
13 ENDSELECT.
```

tablastest

tablastest						
NEW YORK	SAN FRANCISCO	SAN FRANCISCO	NEW YORK	ROME	FRANKFURT	ROME
TOKYO	TOKYO	ROME	ROME	OSAKA	NEW YORK	FRANKFURT
NEW YORK	SAN FRANCISCO	SAN FRANCISCO	NEW YORK	TOKYO	FRANKFURT	FRANKFURT
TOKYO	FRANKFURT	NEW YORK	NEW YORK	FRANKFURT	FRANKFURT	NEW YORK
FRANKFURT	BERLIN	BERLIN	FRANKFURT	SINGAPORE	FRANKFURT	FRANKFURT
SINGAPORE	SINGAPORE	SAN FRANCISCO	SAN FRANCISCO	SINGAPORE	SINGAPORE	JAKARTA
SINGAPORE	TOKYO	FRANKFURT	SAN FRANCISCO	SAN FRANCISCO	SINGAPORE	FRANKFURT
FRANKFURT	FRANKFURT	NEW YORK				NEW YORK

Tarea 2

ABAP ofrece varios tipos de opciones de formato para formatear la salida de programas. Por ejemplo, puede crear una lista que incluya varios elementos en diferentes colores o estilos de formato. La declaración `WRITE` es una declaración de formato que se utiliza para mostrar datos en una pantalla, hay diferentes opciones de formato para la instrucción `WRITE`. La sintaxis de la declaración `WRITE` es:

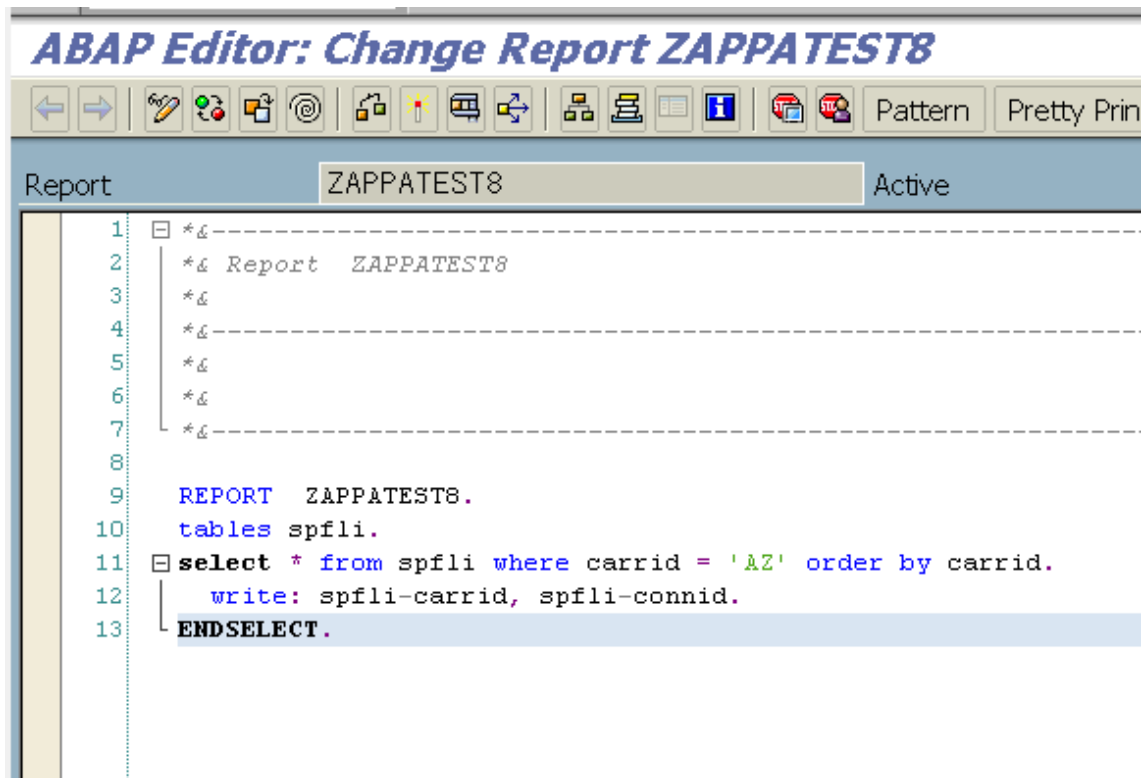
`WRITE <format> <f> <options>.`

En esta sintaxis, `<format>` representa la especificación del formato de salida, que puede ser una barra inclinada (/) que indica la visualización de la salida a partir de una nueva línea además de la barra diagonal, la especificación de formato incluye un número de columna y una longitud de columna.

Disponível em: < <https://isolution.pro/es/t/sap-abap/sap-abap-formatting-data/sap-abap-formateo-de-datos>>. Acesso em: 28 mar. 2023.

Ejercicio 3

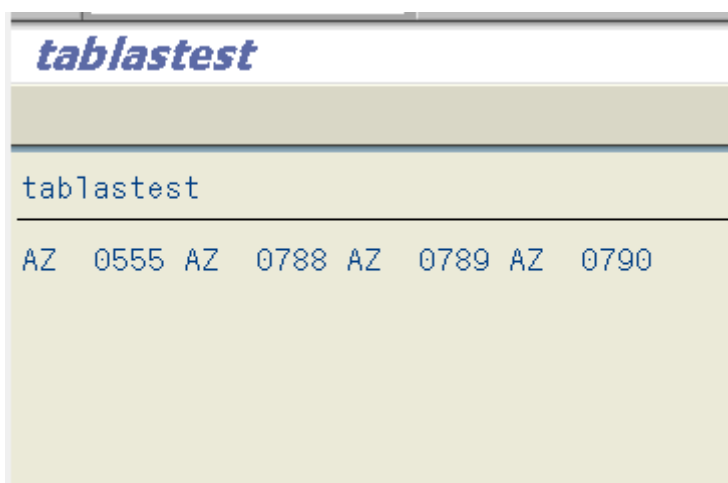
Tarea 1



```
ABAP Editor: Change Report ZAPPATEST8

Report ZAPPATEST8 Active

1  *-----
2  * Report  ZAPPATEST8
3  *
4  *-----
5  *
6  *
7  *-----
8
9  REPORT  ZAPPATEST8.
10 tables spfli.
11 select * from spfli where carrid = 'AZ' order by carrid.
12   write: spfli-carrid, spfli-connid.
13 ENDSELECT.
```



tablastest				
AZ	0555	AZ	0788	AZ 0789 AZ 0790

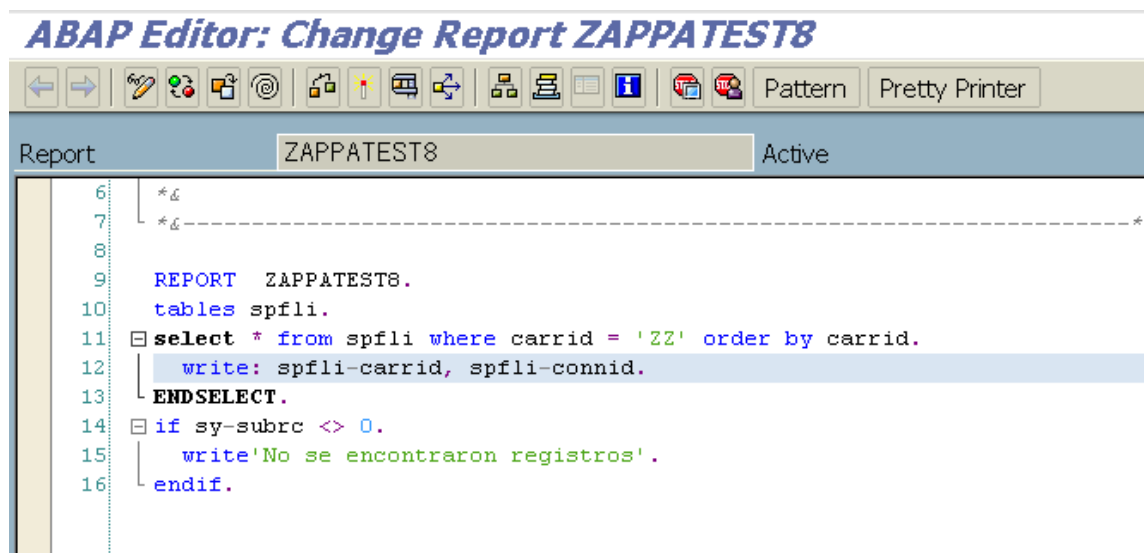
CARRID y CONNID son columnas de tabla utilizadas en la programación ABAP, A través del comando SELECT JOIN, es posible realizar un left outer join de las tablas SCARR y SPFLI utilizando estas columnas

Tiene como salida las variables de sistema que estén disponibles dentro de su programa, uno no tiene que definir nada para acceder a ellas, ellas siempre están disponibles, y se actualizan de forma automática por el sistema como cambios en el entorno de su programa. Todas las variables del sistema comienzan con el prefijo (sy-).

Disponible en: <https://help.sap.com/doc/abapdocu_751_index_htm/7.51/en-us/abapselect_join.htm>. Acceso em: 30 mar. 2023.

Ejercicio 4


Tarea 1



```
ABAP Editor: Change Report ZAPPATEST8

Report ZAPPATEST8 Active

6  *␣
7  *␣-----*
8
9  REPORT ZAPPATEST8.
10 tables spfli.
11 select * from spfli where carrid = 'ZZ' order by carrid.
12   write: spfli-carrid, spfli-connid.
13 ENDSELECT.
14 if sy-subrc <> 0.
15   write 'No se encontraron registros'.
16 endif.
```



```
tablastest

No se encontraron registros
```

En SAP ABAP, el operador "<>" se utiliza para comprobar si dos sentencias son diferentes. Es el equivalente del operador "NE" (no igual) en otros lenguajes de programación, por ejemplo, la condición "IF var1 <> var2" es verdadera si var1 y var2 no tienen el mismo valor.

Disponible en: < https://help.sap.com/doc/abapdocu_731_index_htm/7.31/en-US/abenlogexp_any_operand.htm>. Acceso em: 30 mar. 2023.

Tarea 2

1: En SAP, SY-SUBRC es una variable del sistema que indica el código de retorno de una sentencia ABAP tras su ejecución. Si la sentencia se ejecuta correctamente, el valor devuelto será 0 (cero), en caso contrario el valor será distinto de cero, indicando un error. Cuando la sentencia es una cláusula SELECT, el valor devuelto por SY-SUBRC es 0 si hay registros seleccionados y distinto de 0 si no hay registros seleccionados. Por lo tanto, se recomienda que la comprobación SY-SUBRC se realice después de la cláusula ENDSELECT para asegurarse de que se ha completado la selección.

2: SY-DBCNT es una variable del sistema que contiene el número de filas procesadas en una operación de base de datos. Esta variable es actualizada automáticamente por el sistema y su valor sólo es válido una vez finalizada la operación. En general, se recomienda que la comprobación de SY-DBCNT se realice después de la ejecución de la operación de base de datos, es decir, después de la cláusula END-OF-SELECTION para los informes y después de la cláusula COMMIT WORK para las operaciones de escritura en la base de datos.

Ejercicio 5

Tarea 1

1 Report ztx0505.

2 tables spfli.

3 select * from spfli where carrid = 'ZZ' order by carrid.

4 write: spfli-carrid, spfli-connid.

5 ENDSELECT. 6 if sy-subrc <> 0.

7 write 'No se encontraron registros'. 8 endif.

En este caso, no hay filas coincidentes con los criterios establecidos en el dónde cláusula, por lo que después de la ENDSELECT, el valor de sy-SUBRC se establece en 4. La prueba viene siempre después del ENDSELECT. Si cero filas se devuelven desde la base de datos, el código entre select y ENDSELECT

nunca se ejecuta. Por lo tanto, debe codificar la Para determinar el número de filas devueltas por una declaración select, se debe comprobar el valor de sy-dbcnt después de ENDSELECT. También puede utilizarlo como contador del bucle; entre el select y ENDSELECT, contiene un recuento de la iteración actual. Para la primera fila, SY-dbcnt será 1, para la segunda que será 2, y así sucesivamente. Después de ENDSELECT, conservará su valor, por lo que contendrá el número de filas seleccionadas.

Para determinar el número de filas devueltas por una declaración select , se debe comprobar el valor de sy-dbcnt después de ENDSELECT . También puede utilizarlo como contador del bucle; entre el select y ENDSELECT, contiene un recuento de la iteración actual. Para la primera fila, SY-dbcnt será 1, para la segunda que será 2, y así sucesivamente. Después de ENDSELECT, conservará su valor, por lo que contendrá el número de filas seleccionadas. Por ejemplo:

```
1 Report ztx0506.
2 tables spfli.
3 select * from spfli where carrid = 'AZ' order by carrid.
4 write / sy-dbcnt.
6 write: spfli-carrid, spfli-connid.
7 ENDSELECT.
7 write / sy-dbcnt.
8 write ' registros encontrados'.
```

La línea 4, escribe el valor de sy-dbcnt para cada fila que se devuelve por el select . La barra (/) inicia una nueva línea, por lo que cada sy-dbcnt comienza en una nueva línea.

La línea 5 escribe el valor de los campos en la misma línea que sy-dbcnt Para mostrar los componentes de la estructura sist : Desde el Editor ABAP / 4, pantalla de edición de programas, haga doble clic en el nombre de cualquier campo sy dentro de su código, los dos puntos (:) se llaman el operador cadena se usa para combinar líneas de código que comienzan con la misma palabra o

secuencia de palabras. Coloque la parte común al principio de la declaración seguido de dos puntos.

Disponible en: < <https://answers.sap.com/questions/2859272/what-are-sy-subrc-value-1-2-5-6-means.html>>. Acesso em: 28 mar. 2023.

Ejercicio 6

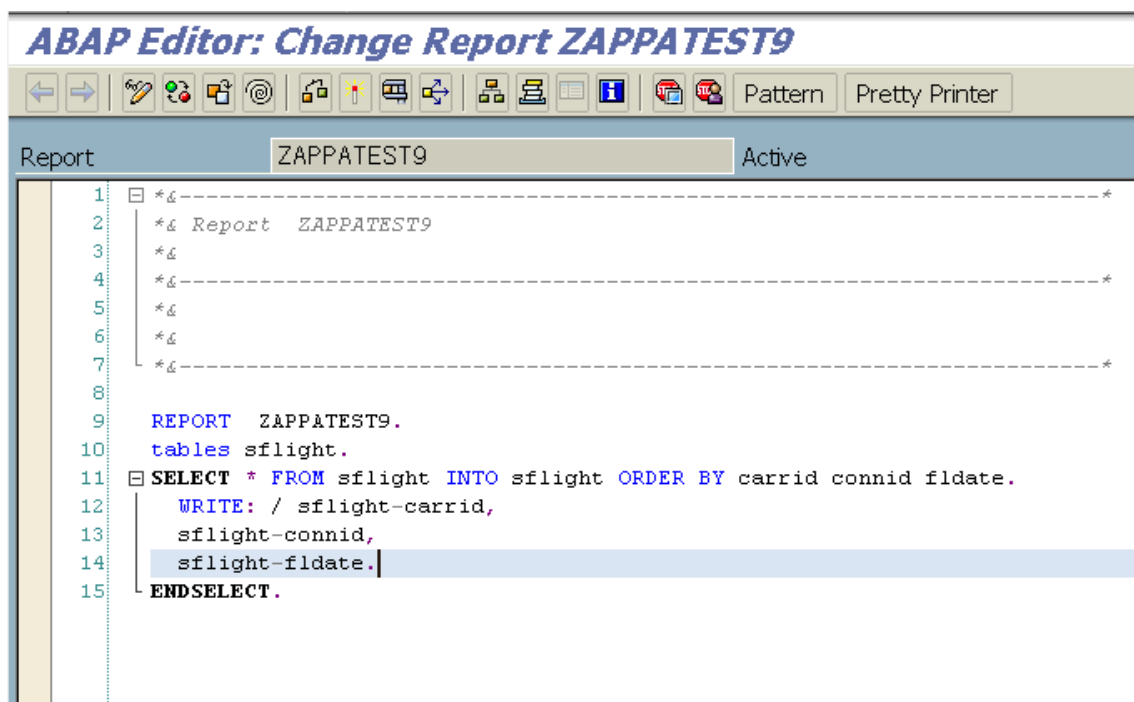
Tarea 1

```
REPORT ZAPPATEST9.

tables sflight.

SELECT * FROM sflight INTO sflight ORDER BY carrid connid fldate.
    WRITE: / sflight-carrid,
           sflight-connid,
           sflight-fldate.

ENDSELECT.
```



Tablatest

Tablatest

AA	0017	26.01.2011	
AA	0017	23.02.2011	
AA	0017	23.03.2011	
AA	0017	20.04.2011	
AA	0017	18.05.2011	AA 0017 25.01.2012
AA	0017	15.06.2011	AA 0017 22.02.2012
AA	0017	13.07.2011	AA 0064 28.01.2011
AA	0017	10.08.2011	AA 0064 25.02.2011
AA	0017	07.09.2011	AA 0064 25.03.2011
AA	0017	05.10.2011	AA 0064 22.04.2011
AA	0017	02.11.2011	AA 0064 20.05.2011
AA	0017	30.11.2011	AA 0064 17.06.2011
AA	0017	28.12.2011	AA 0064 15.07.2011

Ejercicio 7

Tarea 1

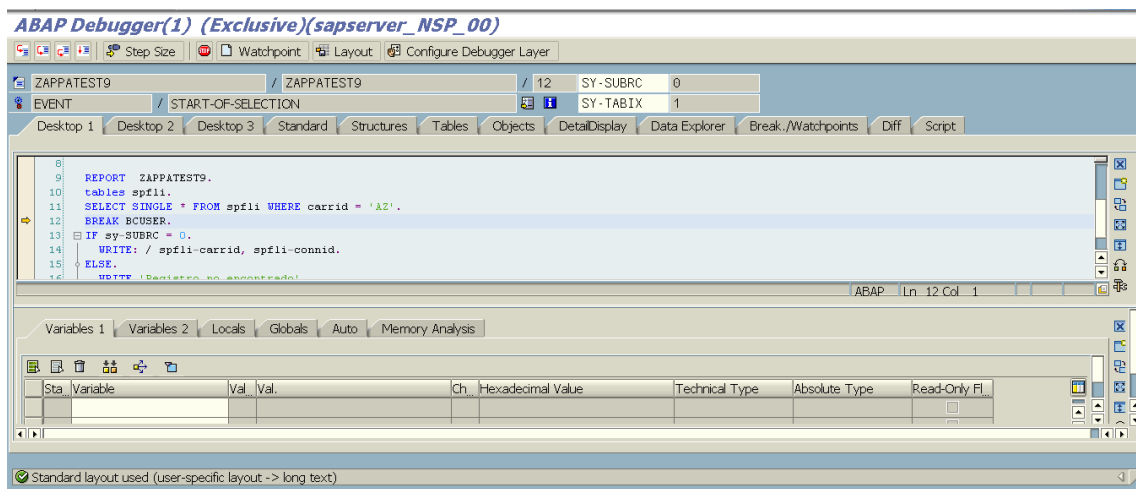
ABAP Editor: Change Report ZAPPATEST8

Report: ZAPPATEST8 Active

```
5  *&
6  *&
7  *&-----
8
9  REPORT  ZAPPATEST8.
10 tables spfli.
11 select single * from spfli where carrid = 'AZ'.
12 if sy-SUBRC = 0.
13     write: / spfli-carrid, spfli-connid.
14 else.
15     write 'Registro no hallado'.
16 endif.
```

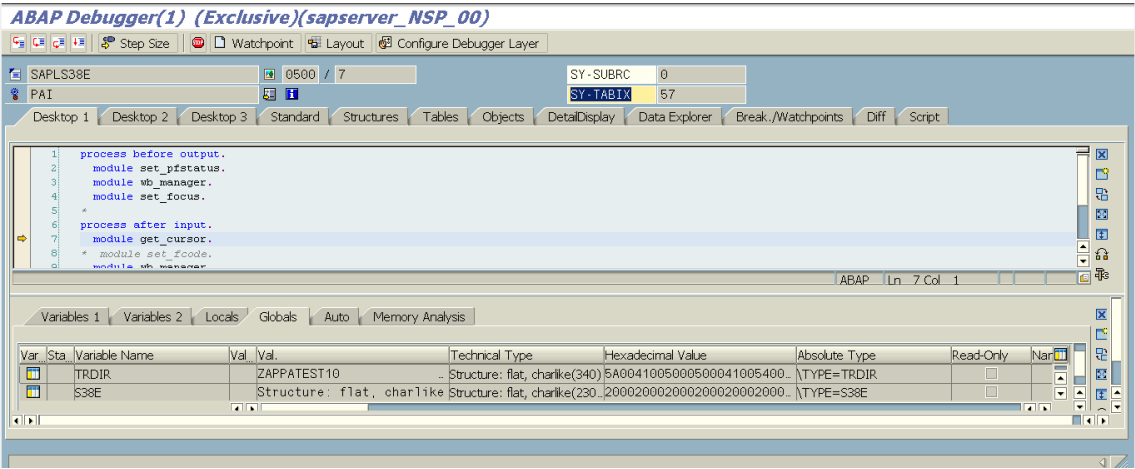
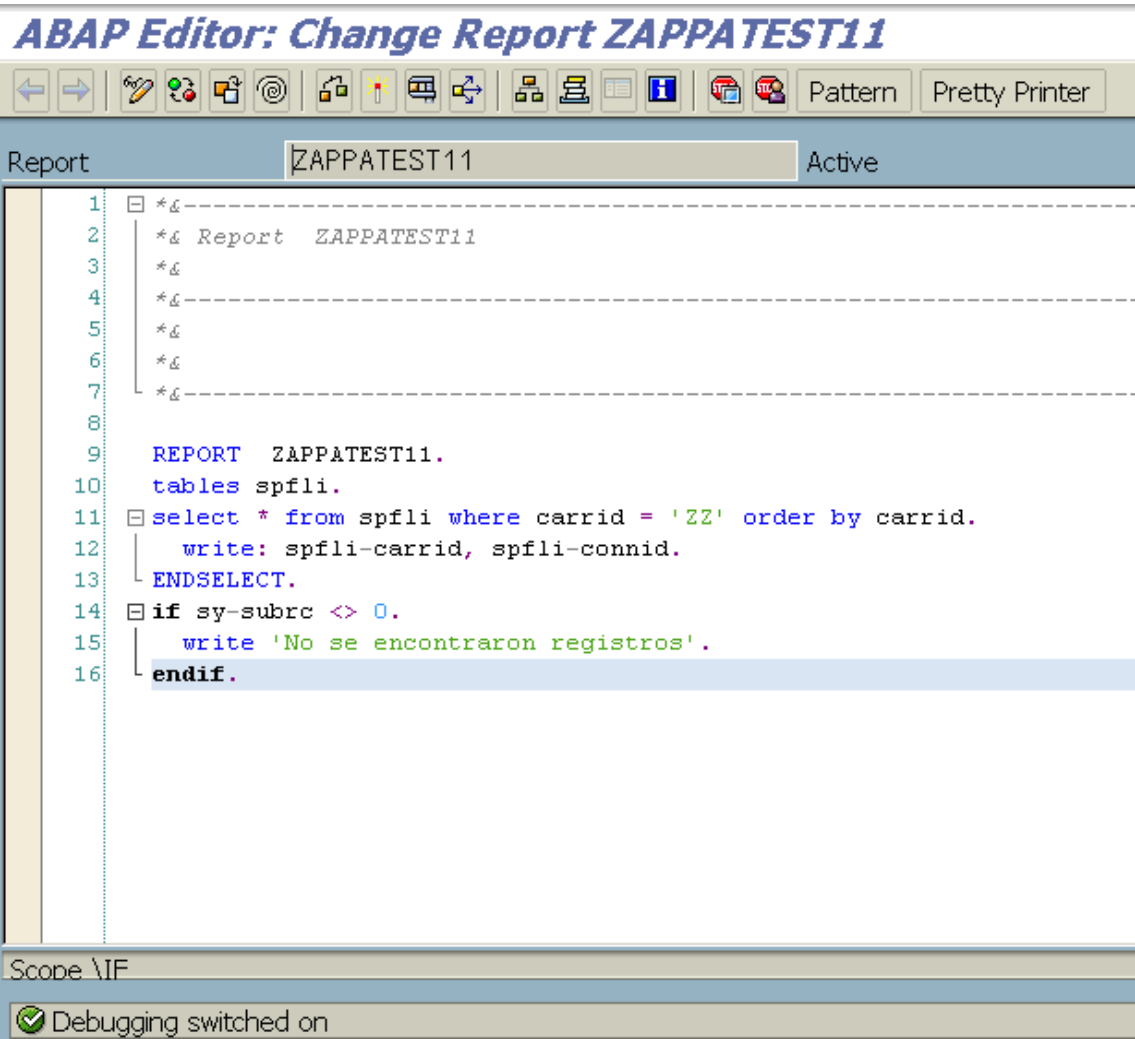


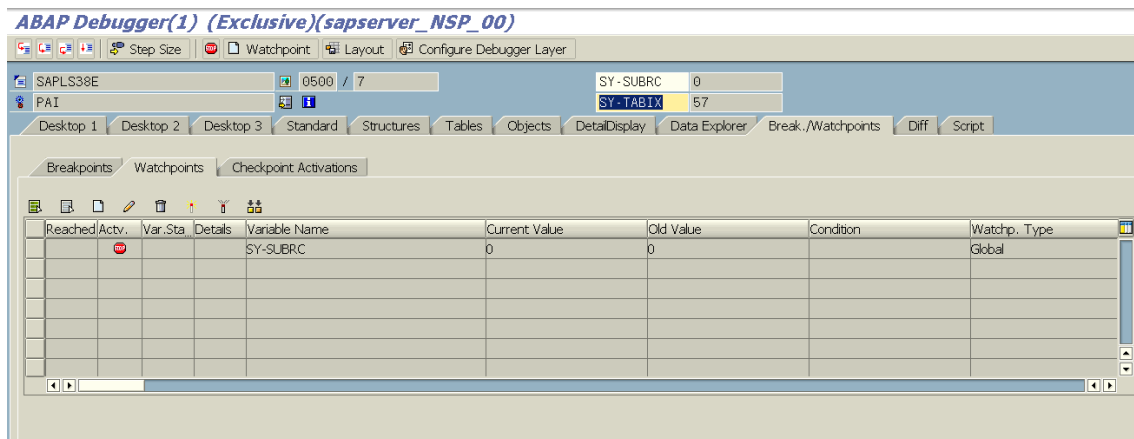

Tarea 2



Ejercicio 8

Tarea 1





Ejercicio 9

Tarea 1

Esta sentencia comprueba si se ha introducido una autorización en el registro maestro del usuario actual o del usuario especificado en user para el objeto de autorización introducido en el campo auth_obj y si esta autorización es suficiente para la solicitud especificada en la sentencia. auth_obj espera un campo similar a un solo carácter que contenga el nombre de un objeto de autorización. Si no se especifica la adición FOR USER, se comprueba la autorización del usuario actual.

Si se especifica id1 ... id10, se deben especificar al menos uno y como máximo 10 campos de autorización diferentes para el objeto de autorización en cuestión. id1 ... id10 espera campos con caracteres simples que contengan los nombres de los campos de autorización en mayúsculas. Si especifica un campo de autorización que no aparece en el objeto de autorización, las comprobaciones no son posibles y sys-subrc se establece en 4. Cada campo de autorización especificado espera un valor que se comprobará mediante FIELD en un campo de tipo carácter plano val1 ... val10 o la adición DUMMY. El valor que se comprueba puede tener un máximo de 40 caracteres y no se evalúan más de 40 posiciones de val1 ... val10. Los campos más largos generan un aviso de comprobación ampliada del programa.

La comprobación de la autorización se realiza si el indicador de comprobación del objeto de autorización especificado para el contexto actual se establece en comprobar con cualquier estado de propuesta. Si el indicador de comprobación se establece como sin comprobación, no se realiza ninguna

comprobación de autorización y sy-subrc se establece en 0 como en una comprobación correcta. La verificación de autorización tiene éxito si se crearon una o varias autorizaciones para el objeto de autorización en el registro maestro de usuario y si, para al menos una de las autorizaciones, cada uno de los ámbitos de valores definidos allí para los campos de autorización especificados utilizando ID incluye el valor val1 ... val10 a comprobar. Los campos de autorización que no se incluyen en la sentencia o que tienen especificado DUMMY no se comprueban. Si la comprobación tiene éxito, sy-subrc se establece en 0. En caso contrario, se le asigna un valor distinto de 0 (véase más adelante).

Ejemplo: Comprueba que el usuario actual tiene autorización para mostrar la línea aérea que ha introducido en la pantalla de selección. El objeto de autorización utilizado aquí es S_CARRID e incluye los campos de autorización CARRID para identificar una línea aérea y ACTVT para la actividad, el código "03" representa la actividad "Visualizar", que es una de las actividades asignadas al objeto de autorización S_CARRID.

PARÁMETROS carr spfli-carrid.

EN LA PANTALLA DE SELECCIÓN.

OBJETO DE AUTORIZACIÓN 'S_CARRID

ID 'CARRID' FIELD carr

ID 'ACTVT' FIELD '03'.

IF sy-subrc <> 0.

MESSAGE 'No authorization' TYPE 'E'.

END IF.

Disponible en: < https://help.sap.com/doc/abapdocu_752_index_htm/7.52/en-US/abapauthority-check.htm>. Acceso em: 31 mar. 2023.

Tarea 2

El mecanismo de autorización de SAP permite que los usuarios tengan permisos específicos en diferentes áreas del sistema, como módulos y submódulos, por ejemplo. Se pueden definir diferentes permisos para cada

usuario o grupo de usuarios, y el sistema dispone de objetos de autorización específicos para comprobar qué permisos son necesarios para realizar una tarea concreta. Además, existen roles asignados a los objetos de autorización, que determinan el nivel de acceso de cada usuario. Por ejemplo, en lo que respecta al SAP Application Interface Framework, el sistema utiliza el objeto de autorización /AIF/CUST para comprobar la autorización del usuario para las actividades de personalización. En lo que respecta al SAP Tax Declaration Framework para Brasil, el objeto de autorización /TMF/R_CONTROL sólo permite la visualización, mientras que /TMF/R_FISCAL permite tanto la visualización como la edición y /TMF/R_CONTAB permite tanto la visualización como la edición. Sin embargo, los permisos de superusuario no pueden modificarse, para saber más sobre cómo funcionan las autorizaciones en SAP Business One 10, hay un manual disponible con la funcionalidad de autorizaciones explicada en detalle .

En cuanto a las autorizaciones específicas para informes, se pueden definir los permisos necesarios para que un usuario pueda ver o generar informes específicos. Sin embargo, los permisos deben configurarse previamente en el sistema y asignarse a los usuarios o grupos de usuarios correspondientes.

Disponible en: https://help.sap.com/doc/saphelp_sbo900/9.0/pt-PT/45/070defe02641dfe10000000a1553f6/content.htm?no_cache=true Acceso em: 31 mar. 2023.

Ejercicio 10

Tarea 1

Tablas internas: La tabla interna es en realidad una tabla temporal, que contiene los registros de un programa ABAP que se está ejecutando. Una tabla interna existe solo durante el tiempo de ejecución de un programa SAP. Se utilizan para procesar grandes volúmenes de datos mediante el lenguaje ABAP. Necesitamos declarar una tabla interna en un programa ABAP cuando necesite recuperar datos de las tablas de la base de datos. Los datos de una tabla interna se almacenan en filas y columnas. Cada fila se llama línea y cada columna se llama field. En una tabla interna, todos los registros tienen la misma estructura y clave.

Se accede a los registros individuales de una tabla interna con un índice o una clave. Como la tabla interna existe hasta que se ejecuta el programa asociado, los registros de la tabla interna se descartan cuando finaliza la ejecución del programa. Por lo tanto, las tablas internas se pueden usar como áreas de almacenamiento o búferes temporales donde los datos se pueden modificar según sea necesario. Estas tablas ocupan memoria solo en tiempo de ejecución y no en el momento de su declaración.

Disponible en: <<https://tutoriales.edu.lat/pub/sap-abap/sap-abap-internal-tables/sap-abap-tablas-internas>>. Acceso em: 31 mar. 2023

Funciones: Los módulos de funciones constituyen una parte importante de un sistema SAP, porque durante años SAP ha modularizado el código utilizando módulos de funciones, lo que permite la reutilización del código, por ellos mismos, sus desarrolladores y también por sus clientes. Los módulos de función son subprogramas que contienen un conjunto de declaraciones reutilizables con parámetros de importación y exportación. A diferencia de los programas de inclusión, los módulos de funciones se pueden ejecutar de forma independiente. El sistema SAP contiene varios módulos de funciones predefinidos que se pueden llamar desde cualquier programa ABAP.

El grupo de funciones actúa como una especie de contenedor para varios módulos de funciones que lógicamente pertenecerían juntos. Por ejemplo, los módulos de funciones para un sistema de nómina de recursos humanos se agruparían en un grupo de funciones. Para ver cómo crear módulos de funciones, se debe explorar el constructor de funciones, puede encontrar el constructor de funciones con el código de transacción SE37.

Disponible en: <<https://tutoriales.edu.lat/pub/sap-abap/sap-abap-function-modules/sap-abap-modulos-de-funciones>>. Acceso em: 31 mar. 2023.

OSS: OSS significa Online Service System y, como su nombre indica, el servicio es accesible a través de Internet. Las notas SAP forman parte del portal de servicios SAP. En términos sencillos, se trata de la base de conocimientos de SAP que aborda específicamente los problemas conocidos del sistema SAP. Se actualiza continuamente a medida que los clientes de SAP informan de

problemas en el sistema SAP y SAP publica una nota con instrucciones detalladas sobre cómo solucionar el problema.

La mayoría de los problemas se solucionan cuando SAP publica la siguiente versión del mismo componente y, por lo tanto, es muy necesario ver la aplicabilidad de una nota con respecto a la versión en la que se está tratando un problema. Cada nota se aplica a una versión y, en la mayoría de los casos, a todas sus versiones predecesoras.

Disponible en: < <https://blogs.sap.com/2013/09/18/sap-oss-notes-an-overview/>>.

Acesso em: 31 mar. 2023.