CHAPTER 1
INTRODUCTION TO ERP

1.1 WHAT IS ERP

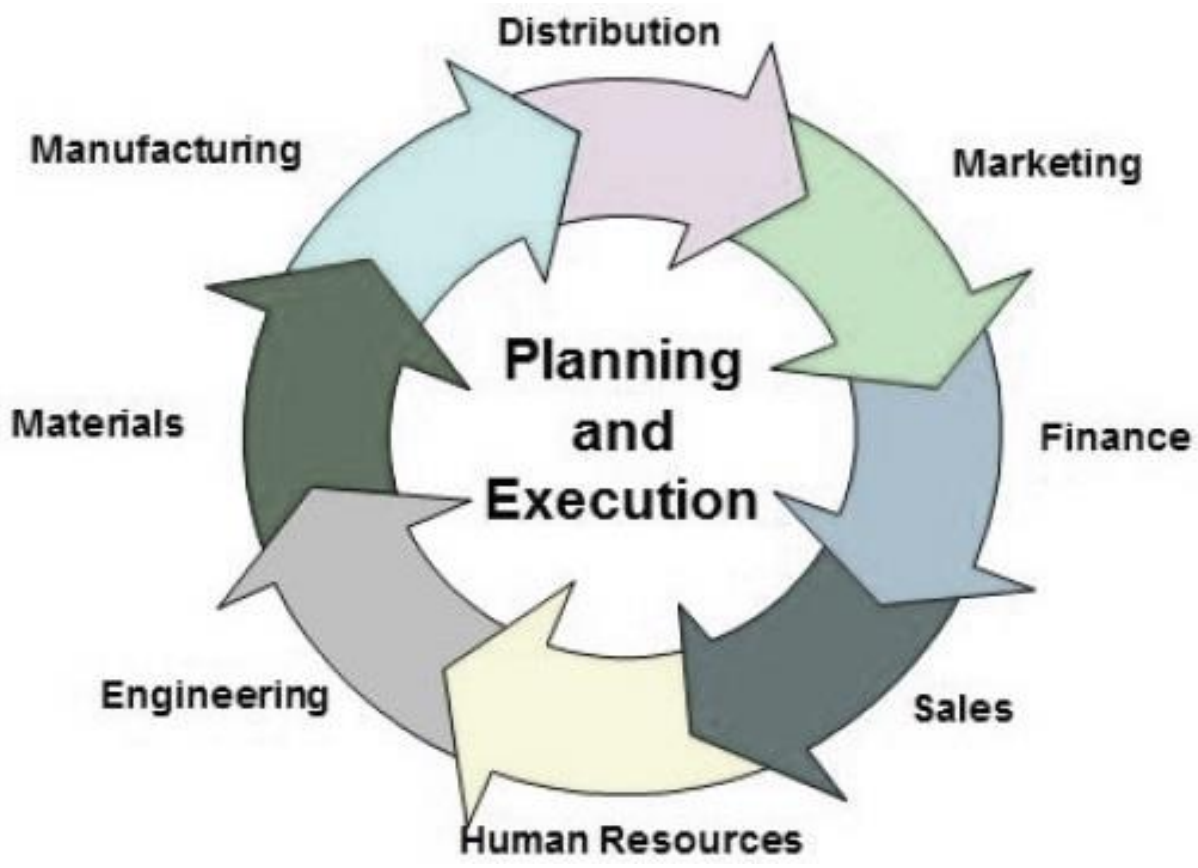
Enterprise resource planning (**ERP**) is the integrated management of core business processes, often in real-time and mediated by software and technology. These business activities can include:

- product planning, purchase
- production planning
- manufacturing or service delivery
- marketing and sales
- materials management
- inventory management
- retail
- shipping and payment
- finance

ERP is usually referred to as a category of business-management software — typically a suite of integrated applications—that an organization can use to collect, store, manage and interpret data from these many business activities.

ERP provides an integrated and continuously updated view of core business processes using common databases maintained by a database management system. ERP systems track business resources—cash, raw materials, production capacity—and the status of business commitments: orders, purchase orders, and payroll. The applications that make up the system share data across various departments (manufacturing, purchasing, sales, accounting, etc.) that provide the data. ERP facilitates information flow between all business functions and manages connections to outside stakeholders.

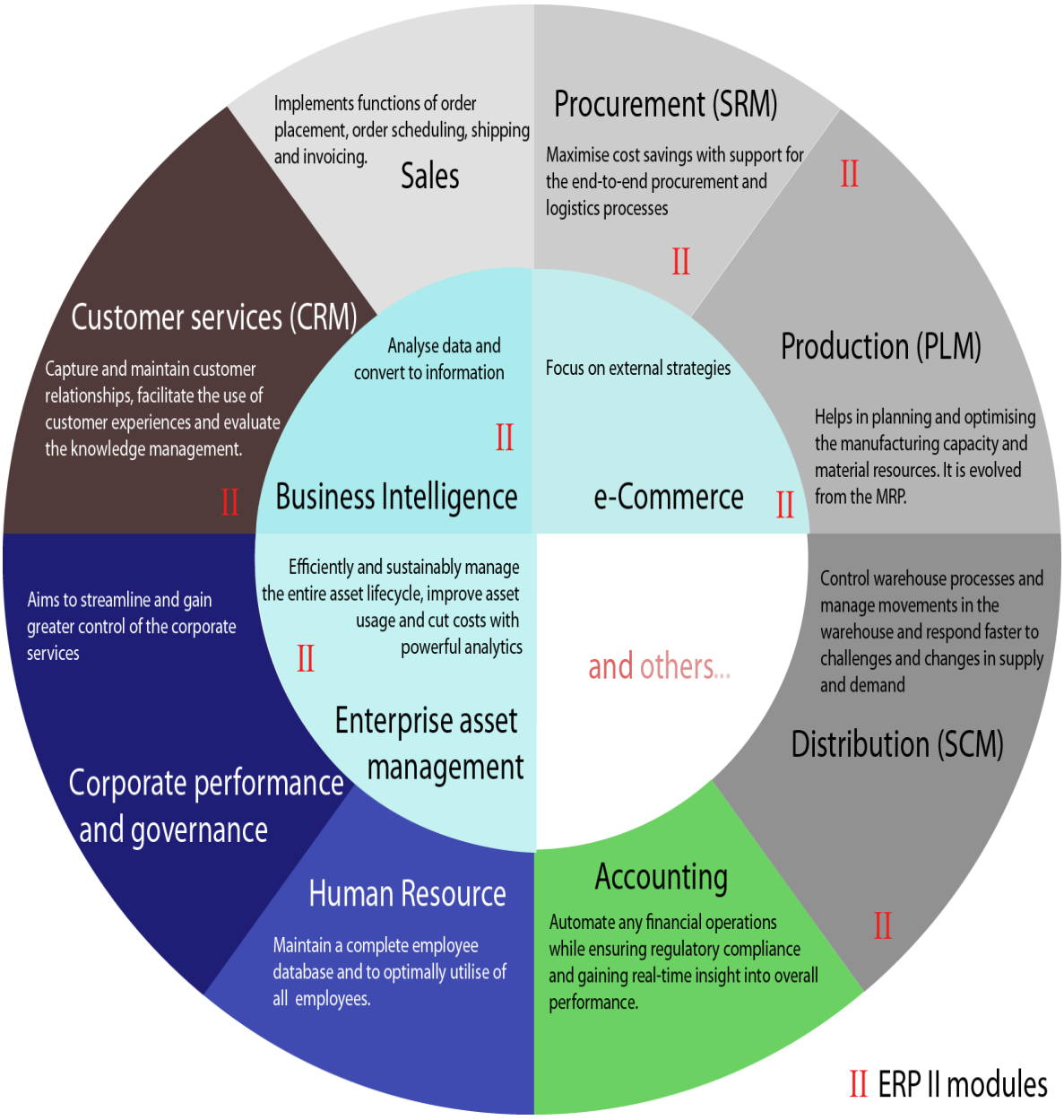
Enterprise system software is a multi-billion-dollar industry that produces components supporting a variety of business functions. IT investments have become the largest category of capital expenditure in United States-based businesses over the past decade. Though early ERP systems focused on large enterprises, smaller enterprises increasingly use ERP systems.



People Soft – emerged with Oracle in 2002

The ERP package is designed to support and integrate almost every functional area of a business process such as procurement of goods and services, sale and distribution, finance, accountings, human resource, manufacturing, production planning, logistics & warehouse management.

1.2 FEATURES AND MODULES



Increase efficiency

Business processes like accounting, sales, marketing, production and inventory are integrated in one ERP platform. It's easier to collect and access data across the organization, streamlining cross-departmental workflows.

Promote collaboration

ERP breaks down walls between departments. Data silos are integrated and a process superhighway links local workstation together. This setup allows teams that used to operate in a vacuum to easily collaborate with other teams inside the ERP platform.

Make accurate forecasts

The only worse thing about the lack of forecast is a wrong one. Forecasts shape strategies; thus, it's crucial organizations get the real picture. Using a centralized database, ERP lends to a company's disparate business solutions a standardized process, ultimately, enhancing data integrity.

ERP reporting tools use advanced filters and analytics to sift data for inconsistencies. Features of ERP software like reduplication also ensure data is updated and duplicate-free.

Lower operational costs

A company can also leverage ERP to cut down costs. When processes are streamlined and key metrics are closely monitored, disruptions, delays and breakdowns are anticipated .

Increase data security

ERP solutions have firewalls and restriction controls to guard against data breach. Having a single data warehouse means access points are tightly monitored and security is concentrated.

Comply with regulations

Many ERP solutions feature built-in regulatory process standards and compliance reporting to help businesses meet myriad business requirements. ERP solutions subscribe to reporting protocols for aspects like financial accounting, product regulations and data security.

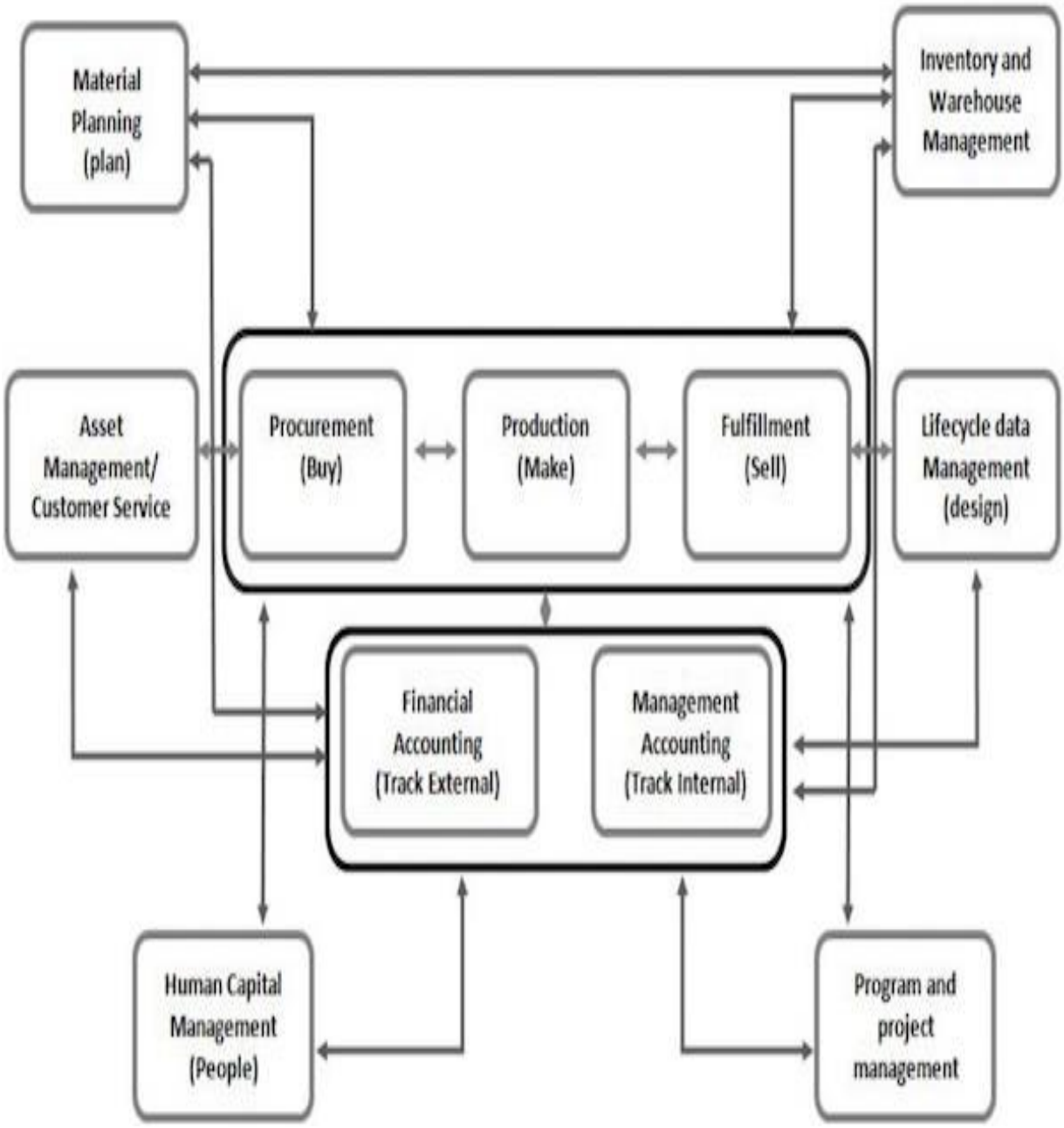
1.3 BUSINESS PROCESS INTEGRATION

Every business, regardless of the industry they belong to, require connected systems with efficient information flow from one business process to another. Business Process Integration (BPI) plays an important role in overcoming integrating challenges that allows organizations to connect systems internally and externally.

Business Process Integration (BPI) allows −

- automation of business processes,
- integration of systems and services,
- secure sharing of data across numerous applications, and
- automation of management, operational, and supporting process.

The following illustration shows an overview of various business processes running in an enterprise and how they are integrated.



1.4 EVOLUTION OF ERP

During early phases of development, integrated solutions were designed for particular process areas such as −

- Material Management − the integrated system was known as Material Requirement Planning (MRP)
- Manufacturing − the integrated system was known as Manufacturing Resource Planning

However none of the integrated systems came with a complete solution for an organization covering major business process areas. In early 1990's, the Gartner Group first used the acronym **ERP**. By mid–1990's, ERP systems addressed all the core enterprise functions.

In the early stages, most of the ERP solutions were focused on automating back office functions that were not directly affecting customers or general public. Later, front office functions such as customer relationship management and e-business systems were integrated.

1.4 FUNCTIONS OF ERP

An ERP system typically performs the following functions −

- Supports the integrated business process inside the organization.
- Improves capital planning and helps in executing organizational plans and strategies.
- Helps speed up the decision-making process over the analysis of accurate data.
- Helps extend the business network to wider domains, expanding the products and services to reach more customers, suppliers, and partners.
- Identifies operational risks to improve governance.
- Provides protection against organizational data breaches and security threats to leakage of information.
- Makes the organization adaptable to the rapid changes in the business process according to the needs.
- Gives long-term profit by providing means to increase the customer base.

1.5 FUNCTIONAL AREAS

ERP is a business management software is usually a suite of integrated applications that a company can use to collect, store, manage, and interpret data from many functional areas including −

- **Financial Accounting** − Deals with financial transactions and data.
- **Human Resource** − Deals with information related to employee of an organization.
- **Customer Relationship Management** − Deals with capturing and managing customer's relationship, facilitating the use of customer experience to evaluate the knowledge database.
- **Sales and Distribution** − Deals with order placement, delivery, shipment and invoicing.
- **Logistics and Warehouse Management** − Deals with storage of products and shipment.
- **Manufacturing and Material Management** − Deals with the production and production planning activities.
- **Supply Change Management** − Deals with the movement of products, storing, managing, and controlling supplies.
- **Business Intelligence** − Analyzes data and converts the same to information.

1.6 ADVANTAGES OF ERP

By integrating the business processes, the ERP offers the following advantages-

- Saves time and expenses.
- Allows faster decision-making by the management, utilizing the data and reporting tools designed in the systems.
- Single data source and sharing of data among all the units of an organization.
- Helps in tracking every transaction that takes place in an organization, from starting till end.
- Supplies real-time information whenever required.

- Provides synchronized information transfer in between different functional areas such as sales, marketing, finance, manufacturing, human resource, logistics, etc.

1.7 DISADVANTAGES OF ERP

It is not always easy to incorporate ERP in an organization. ERP suffers from the following drawbacks

- Sometimes business processes critical to an organization are to be re-engineered to align them with an ERP solution.
- Cost of complex integration can be very high.
- Switching from one ERP solution to another increases the implementation cost even further.
- End-users are to be trained for their daily operations.
- Customization is not preferred.

CHAPTER 2. INTRODUCTION TO SAP

2.1 HISTORY OF SAP

What is SAP?

SAP stands for **S**ystems **A**pplications and **P**roducts in Data Processing.

SAP by definition is also named of the ERP (Enterprise Resource Planning) software as well the name of the company.

- SAP Software was founded in 1972 by Wellenreuther, Hopp, Hector, Plattner and Tschira.

- SAP system consists of a number of fully integrated modules, which covers virtually every aspect of the business management.

- SAP is first in the ERP market. As of 2010, SAP has more than 140,000 installations worldwide, over 25 industry-specific business solutions and more than 75,000 customers in 120 countries.

- Other Competitive products of SAP Software in the market are Oracle, Microsoft Dynamics etc.

2.2 FUNCTIONAL MODULES OF SAP

SAP solutions include a number of functional modules, which support transactions to execute key business processes, such as −

- Financial Accounting (FI)

- Financial Supply Chain Management (FSCM)
- Controlling (CO)
- Materials Management (MM)
- Sales and Distribution (SD)
- Logistics Execution (LE)
- Production Planning (PP)
- Quality Management (QM)
- Plant Maintenance (PM)
- Project System (PS)
- Human Resources (HR)

2.3 WHY SAP?

- Tightly integrated
- 70+ languages
- 70+ modules
- Real time information
- 28+ verticals (different domains)

CHAPTER 3: ARCHITECTURE OF SAP R/3 ARCHITECTURE

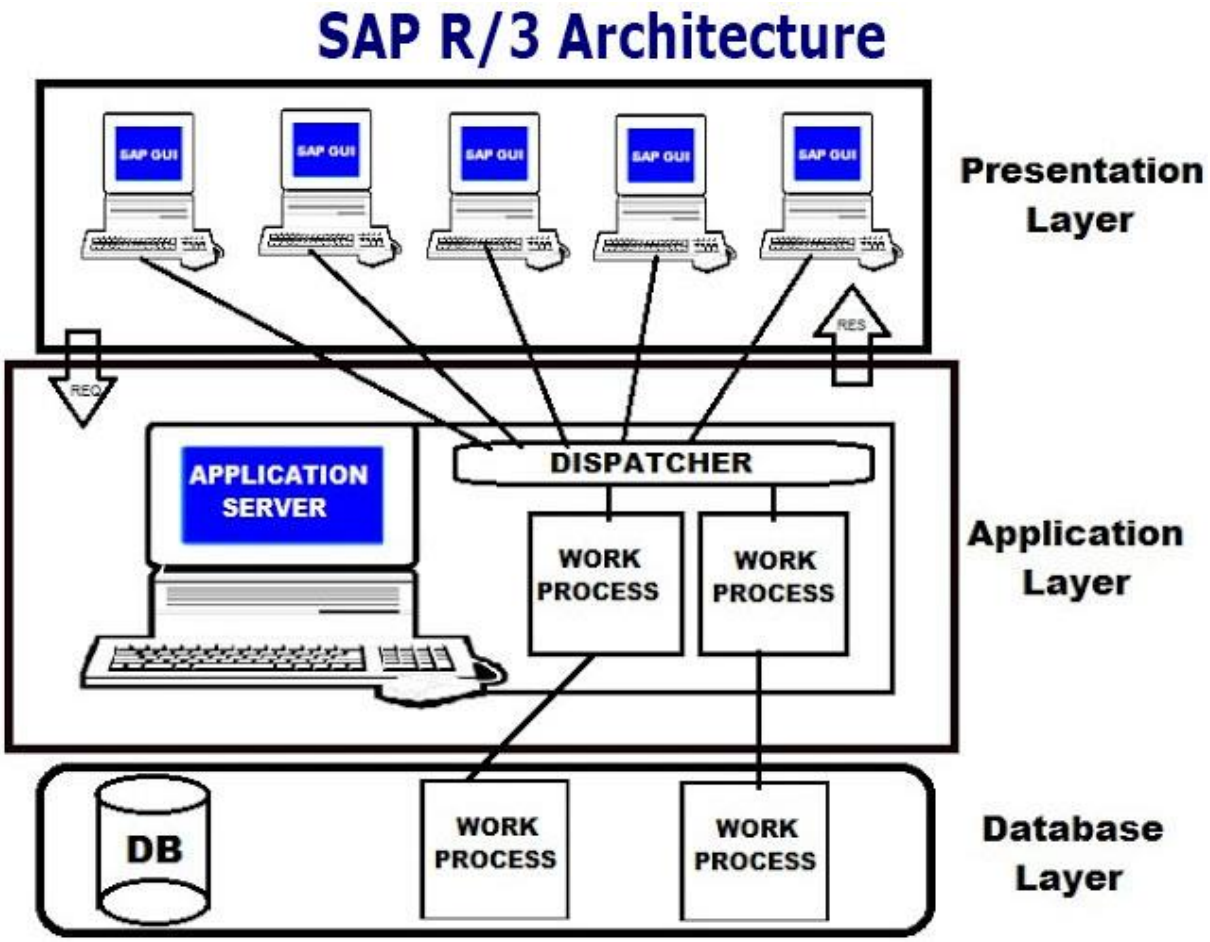
3.1 WHAT IS SAP ECC R/3?

SAP R/3 is a 3 tier architecture consisting of 3 layers

1. Presentation
2. Application
3. Database

In simple words, it's client server architecture.

- **R** signifies Real-time system
- **3** represents - 3-tier architecture.



3.2 T. CODES

TRANSACTION CODE

The Transaction code starts a screen sequence. We can create the transaction code in the Repository Browser in the ABAP Workbench or by using the transaction code SE93. A transaction code is linked to an ABAP program and an initial screen. We can start a screen sequence from any ABAP program using the CALL SCREEN statement.

- S001 ABAP Development Workbench
- S002 System Administration.
- SA38 Execute a program.
- SE11 ABAP/4 Dictionary.
- SE12 Dictionary: Initial Screen – enter object name

- SE13 Access tables in ABAP/4 Dictionary.
- SE14 ABAP/4 Dictionary: Database Utility.
- SE14 Utilities for Dictionary Tables
- SE51 Screen Painter: Initial Screen.
- SE37 ABAP/4 Function Modules
- SE38 ABAP Editor
- SE16 Data Browser
- SE16n Display table contents

CHAPTER 4. ABAP TOOL

4.1 WHAT DOES ADVANCED BUSINESS APPLICATION PROGRAMMING (ABAP) MEAN?

- Advanced business application programming (ABAP) is a 4GL application-specific programming language developed in the 1980s by the German software company SAP. The syntax of ABAP is somewhat similar to COBOL.
- ABAP was and remains the programming language for the development and modification of SAP applications.

- The widely installed R/3 system was first released by SAP in 1992 and developed in ABAP.

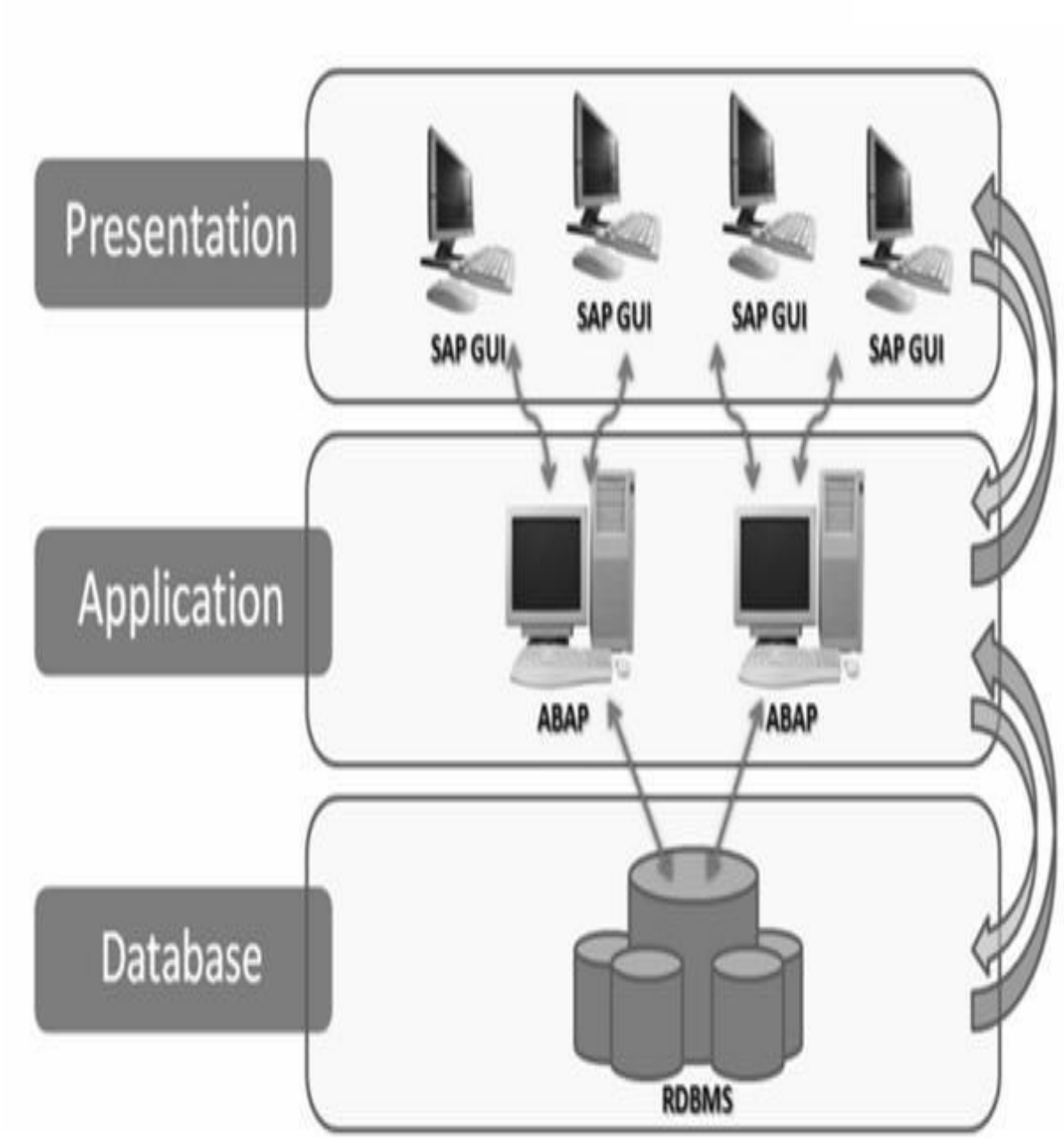
- In 1999, SAP released an object oriented extension to ABAP, which denoted ABAP objects. In 2004, SAP introduced its current development environment called Net Weaver, which supports both ABAP and Java.

- Translated from German, ABAP stands for Allgemeiner Bericht saufbereitungs prozessor which means “generic report preparation process.”

- SAP ABAP is a high-level language that is primarily used to develop enterprise application for large business and financial institution on SAP platform.

- The 3-tier Client/Server architecture of a typical SAP system is depicted as follows:

3-Tier Client/Server Architecture



4.2 MODULE POOL PROGRAMMING IN ABAP

A module pool is a collection of screens, flow logic, menu bars and ABAP code that we use to build an application. When we use the transaction code to call the application it will refer to the module pool and the initial screen number.

Module Pool Programming is a special type of programming which is used to create custom SAP screens. The Transaction code used for creating module pool programs is SE80.Every module pool program must be executed with a Transaction code. Module Pool Programming is also known as ABAP Transaction (or) Dialog Programming (or) User Interface Programming. While working with the module pool programming, user can create new applications as well as modify the existing applications. To work with the module pool programming, the user need to follow the Transaction Code SE38 which allows access to the ABAP Editor Program Type (P.G.Type).

Dialog Programming is used when the ABAP program demands the user input. The user dialog is any form of the interaction between the user and the program which could be data entry, choosing a menu item, clicking button, clicking or double clicking an entry. This dialog program is also used when we need to navigate back and forth between the screens. Dialog programs are created with type as ‘M’ – Module Pool. They cannot be executed independently and must be attached to at least one transaction in which we specify an initial screen.

Dialog Programming allows us to work interactively with the system and to change the contents of the database tables. Each dialog program has a certain sequence of screens that are processed by the system one after the other. It helps to link all the objects hierarchically to the main program, executed in a sequence dictated by the dialog main program.

4.3 COMPONENTS OF THE DIALOG PROGRAM:

- Transaction Code
- Screens
- GUI Status
- ABAP Program
- Screen Flow Logic
- Dynamic Program
- ABAP Module Pool

Transaction code

The Transaction code starts a screen sequence. We can create the transaction code in the Repository Browser in the ABAP Workbench or by using the transaction code SE93. A transaction code is linked to an ABAP program and an initial screen. We can start a screen sequence from any ABAP program using the CALL SCREEN statement.

Screens

Each dialog in an SAP system is controlled by one or more screens. We can create the screens using the Screen Painter in the ABAP Workbench through the transaction code SE51. Each screen belongs to an ABAP program. These screens consist of a “layout” and flow logic. The layout determines the positions of the input/output fields and the other graphical elements such as checkboxes and radio buttons. The flow logic determines the logical processing within the screen.

GUI status

Each screen has GUI status (es) which are independent components of a program. This controls the menu bars, standard toolbar, application toolbar with which the user can choose functions in the application and can be created using the Menu Painter in the ABAP Workbench.

The toolset:

Central Component	Tool	Transaction
All Components	Object Browser	SE80 Tools > ABAP Workbench > Object Browser
Screen	Screen Painter	SE51 Tools > ABAP Workbench > Screen Painter
ABAP/4 Module Pool	ABAP/4 Editor	SE38 Tools > ABAP Workbench > ABAP/4 Editor
Dictionary Objects (tables, fields, etc.)	ABAP/4 Dictionary	SE11 Tools > ABAP Workbench > ABAP/4 Dictionary
Menu	Menu Painter	SE41 Tools > ABAP Workbench > Menu Painter
Transaction	Maintain Transaction	SE93 Tools > ABAP Workbench > Development > Other Tools > Transactions

ABAP Program

Each screen and GUI status in the R/3 system belongs to one ABAP program. ABAP programs are also known as Dialog Programs. The ABAP program contains the dialog modules which are called by the screen flow logic and also process the user input from the GUI status. In a module pool (M type program) the first processing block is always a dialog module. We can also use screens in the other ABAP programs (executable programs or function modules).

Screen Flow Logic

Screen Flow Logic consists of events that are involved in the Module Pool Programming:

Process before Output (**PBO**):

This event is processed before the screen is displayed. This event is triggered before displaying the output of the program.

Process after Input (**PAI**):

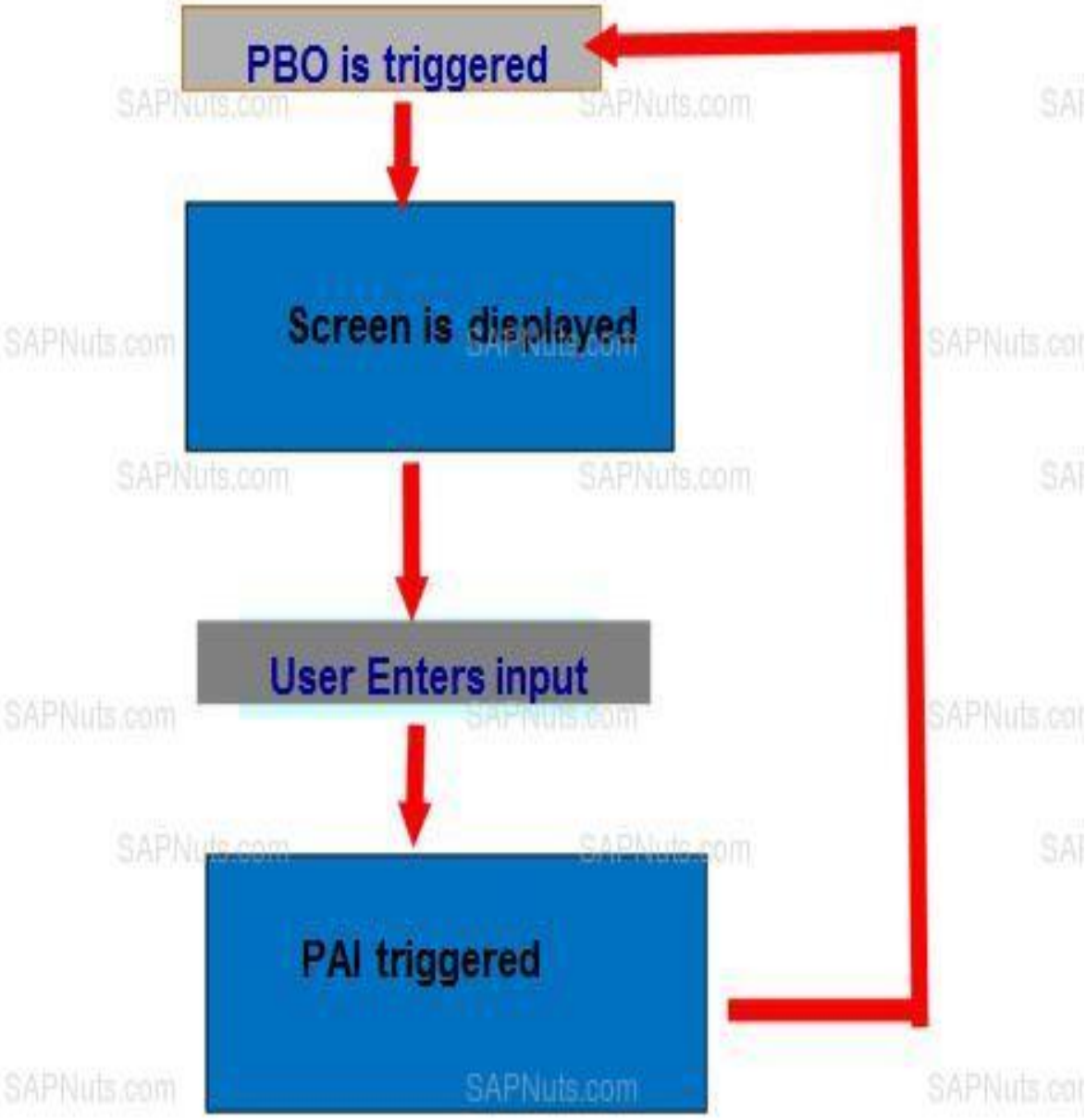
This event is processed after a user action on the screen. This event gets triggered after giving the input for the program.

Process on value request (**POV**):

This event is processed on a valid request or when F4 is pressed. It is used to customize the search help.

Process on help request (**POH**):

This event is processed on a help request or when F1 is pressed. It is used to customize the documentation.



Dynamic Program

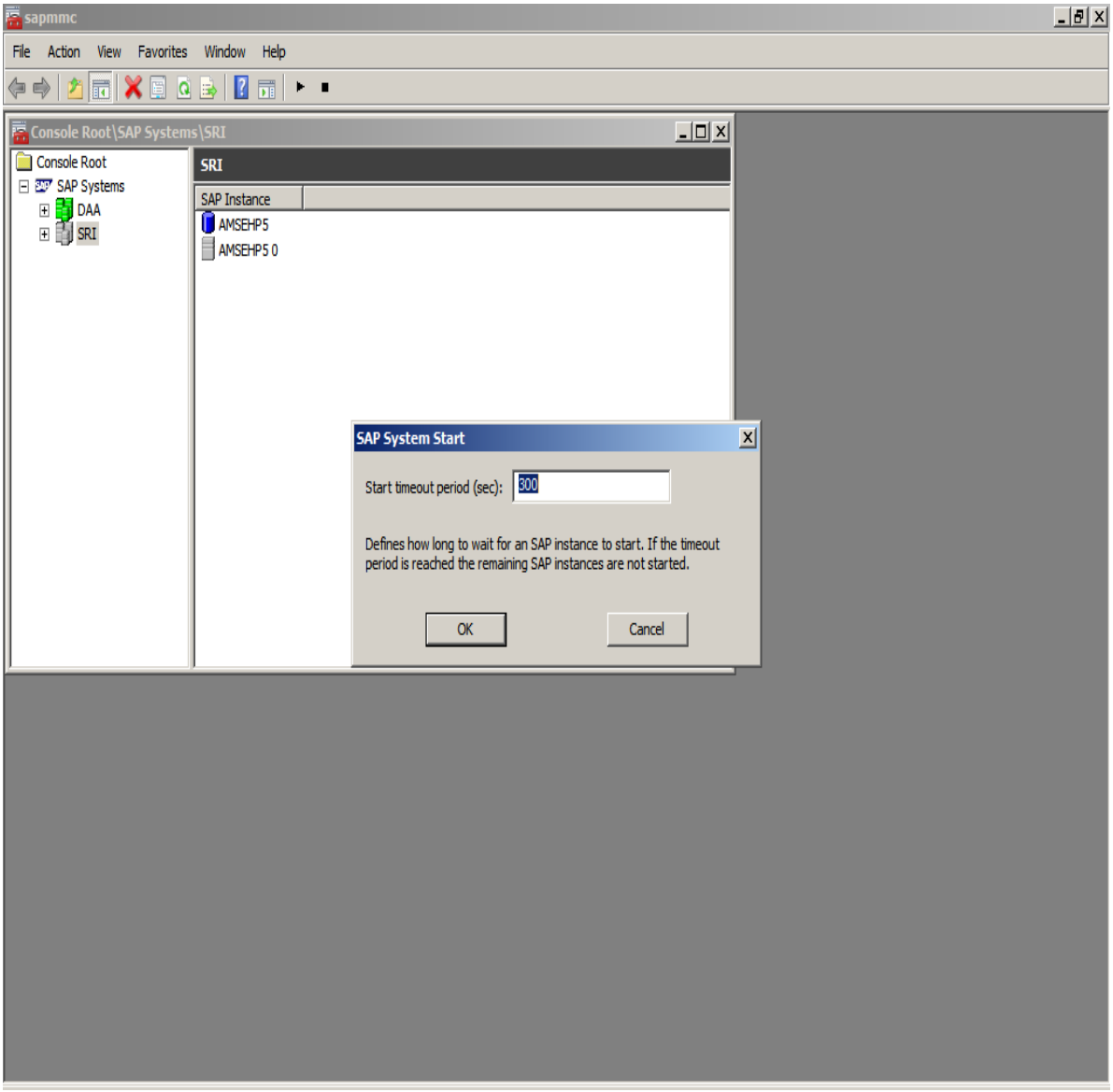
Dynamic Program is defined as a screen together with its flow logic. It controls exactly one step of the Dialog Program. The screens are numbered which are belonging to the program. The screen flow sequence can be either linear or cyclic. From a screen chain we can call another screen chain and after processing it we can return to the original screen chain. Here we can also override the statically-defined next screen from within the dialog modules of the ABAP progra

CHAPTER 5

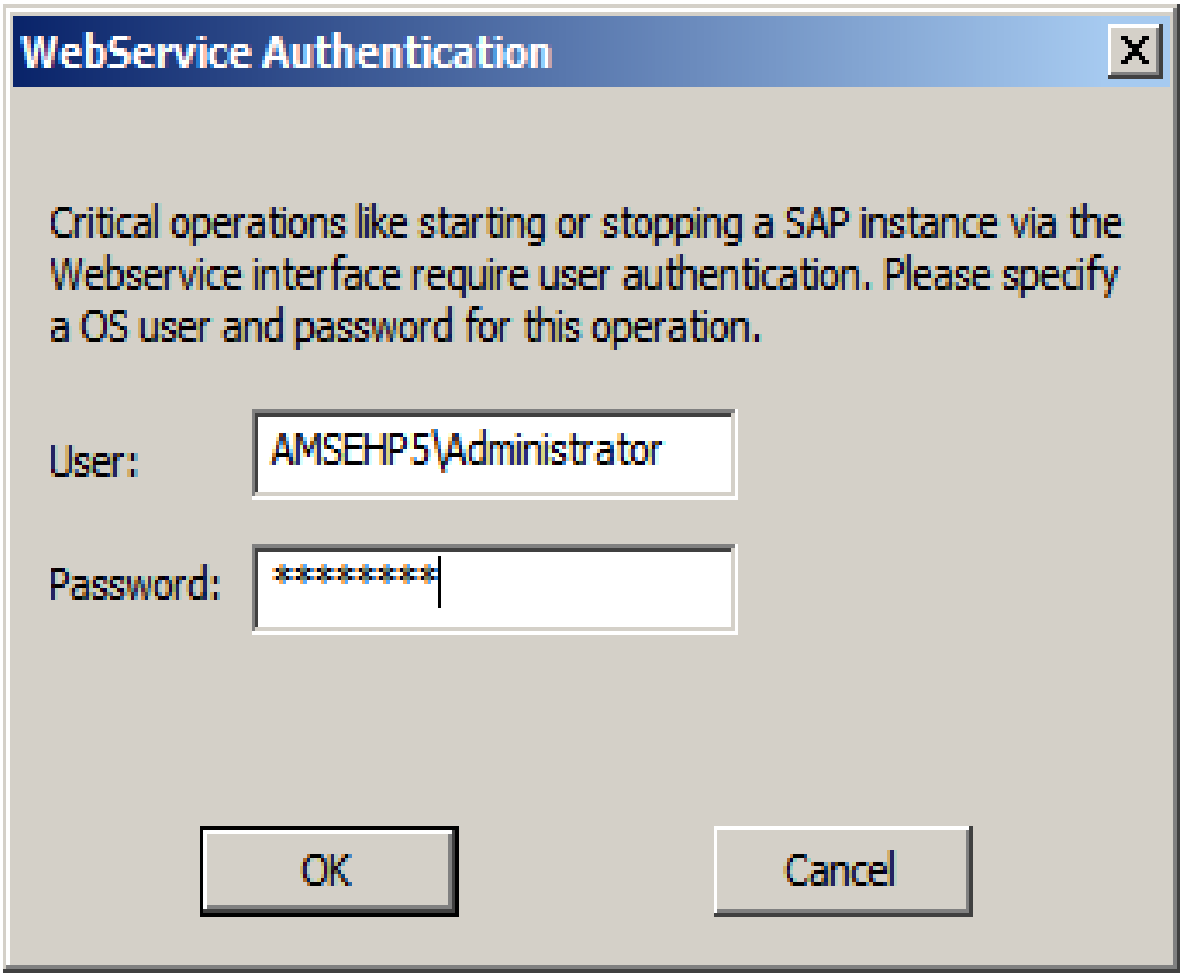
WORKING WITH SAP

5.1 START SAP MANAGEMENT CONSOLE

To start working with SAP we need to start the SAP Management console. Click on SRI and right click on that you will find a drop down and select start. Then following window will appear and click on ok.

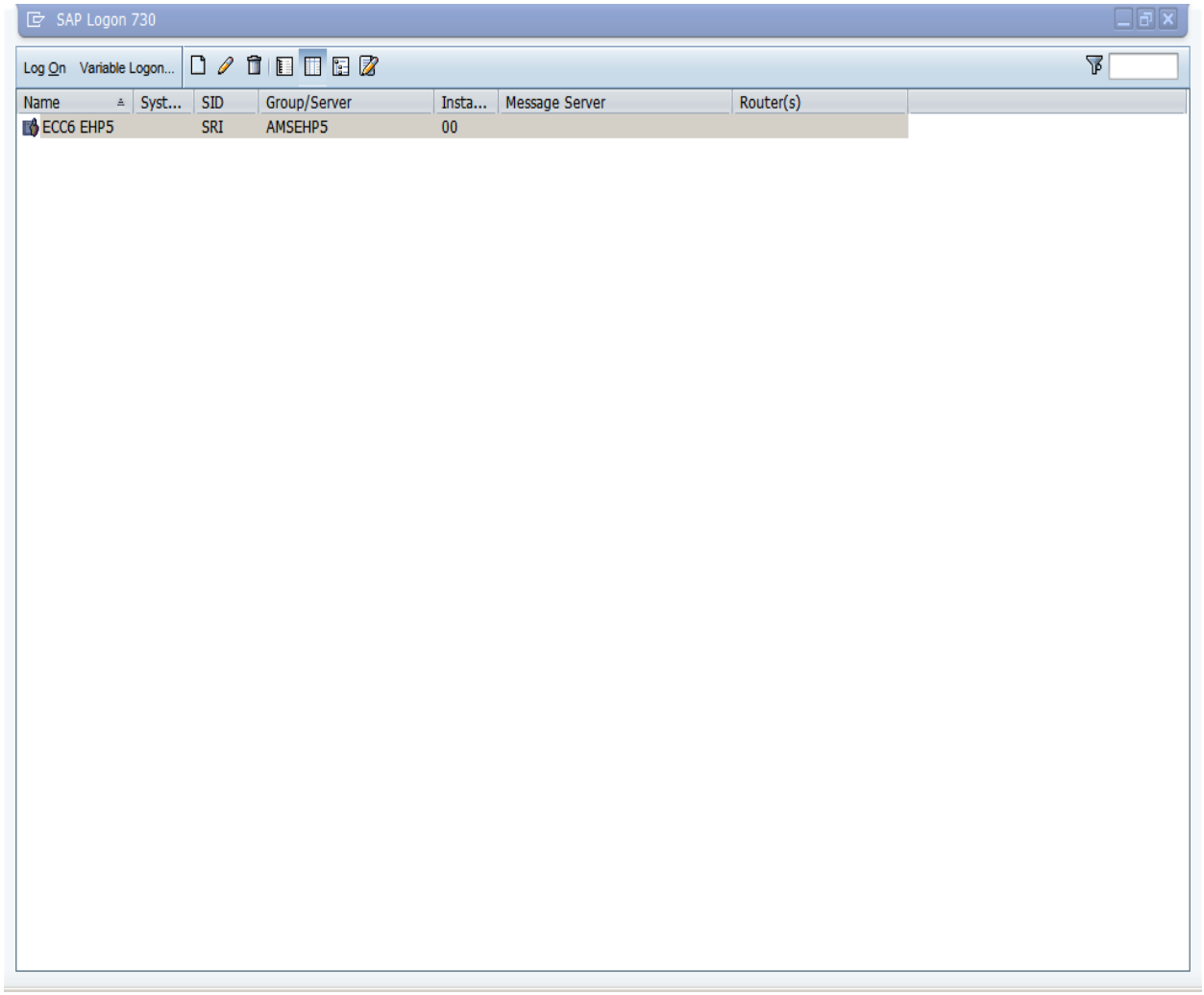


After clicking ok button the following window appears:



Enter the password which is set while installing and then click on ok.

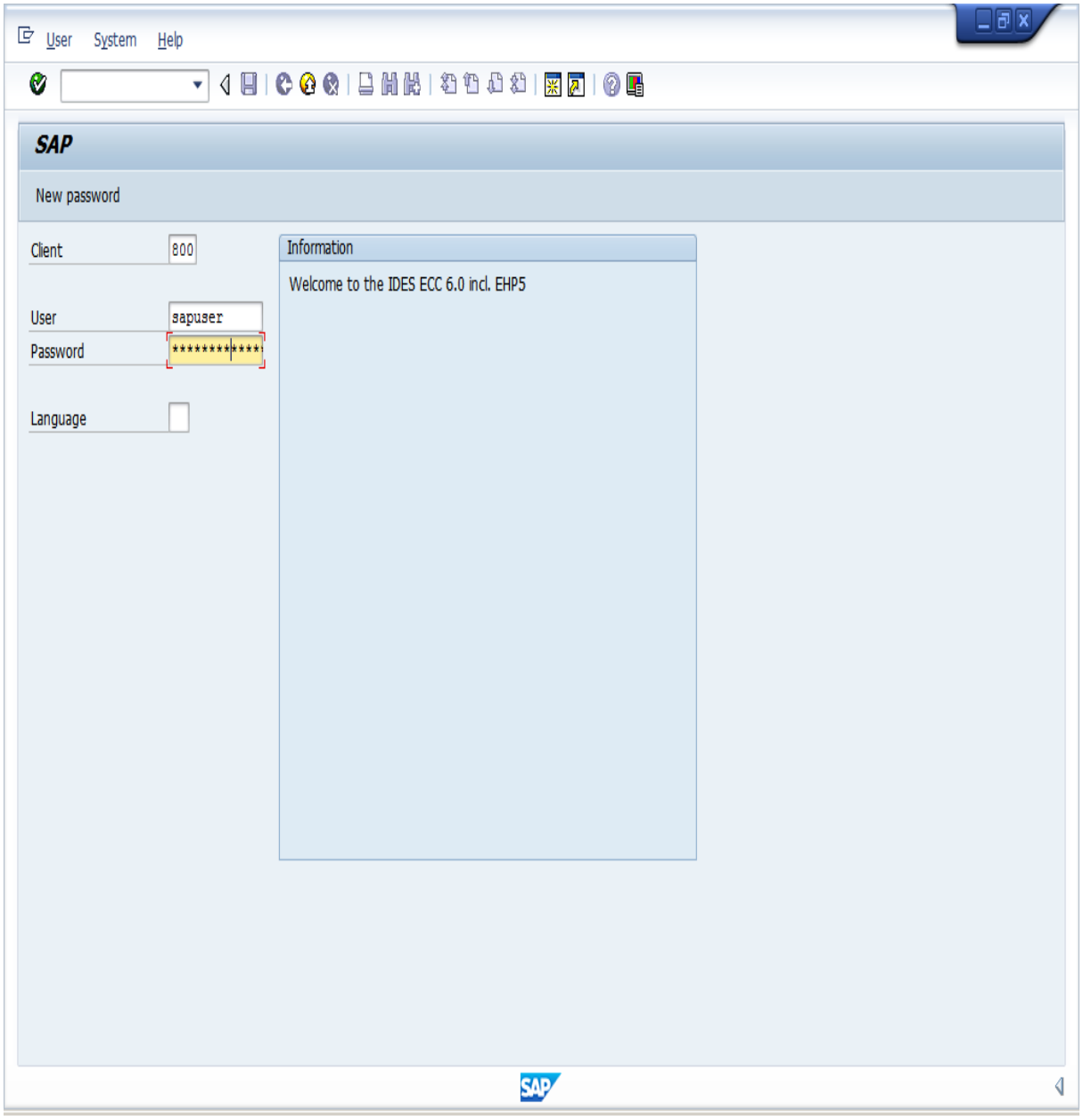
Now you can logon with SAP by click on SAP logon icon.



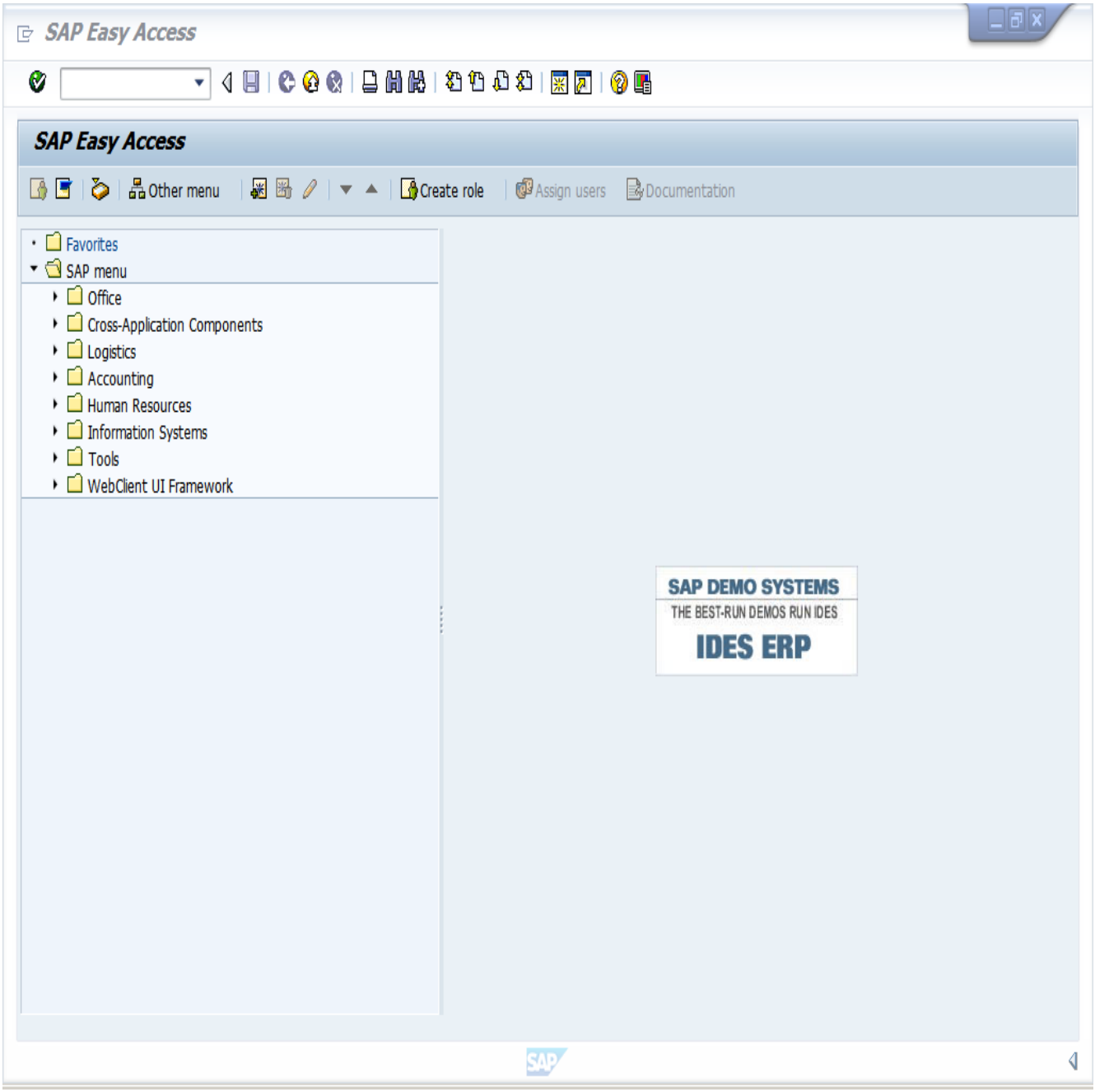
At the top left corner we see the log on icon. Click on it we will find the below screen as follows .

5.2 SAP LOGON

Click on Log on which is at top left corner. You will find below window.



Enter username and password then you will be navigated to below page.



5.3 PACKAGE CREATION

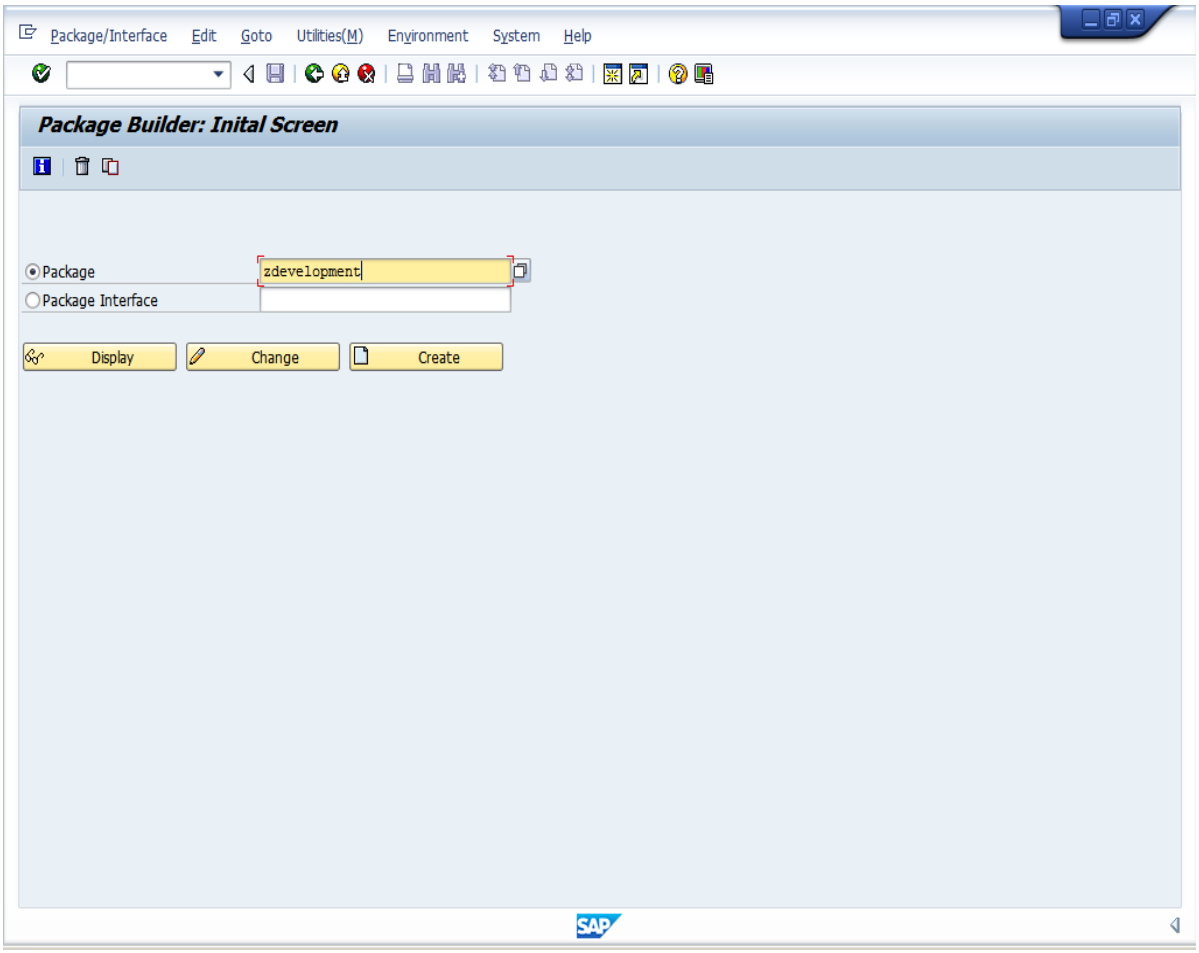
Package can be defined as a collection of folders.

In production environment, package has been created by SAP BASIS consultant in real time.

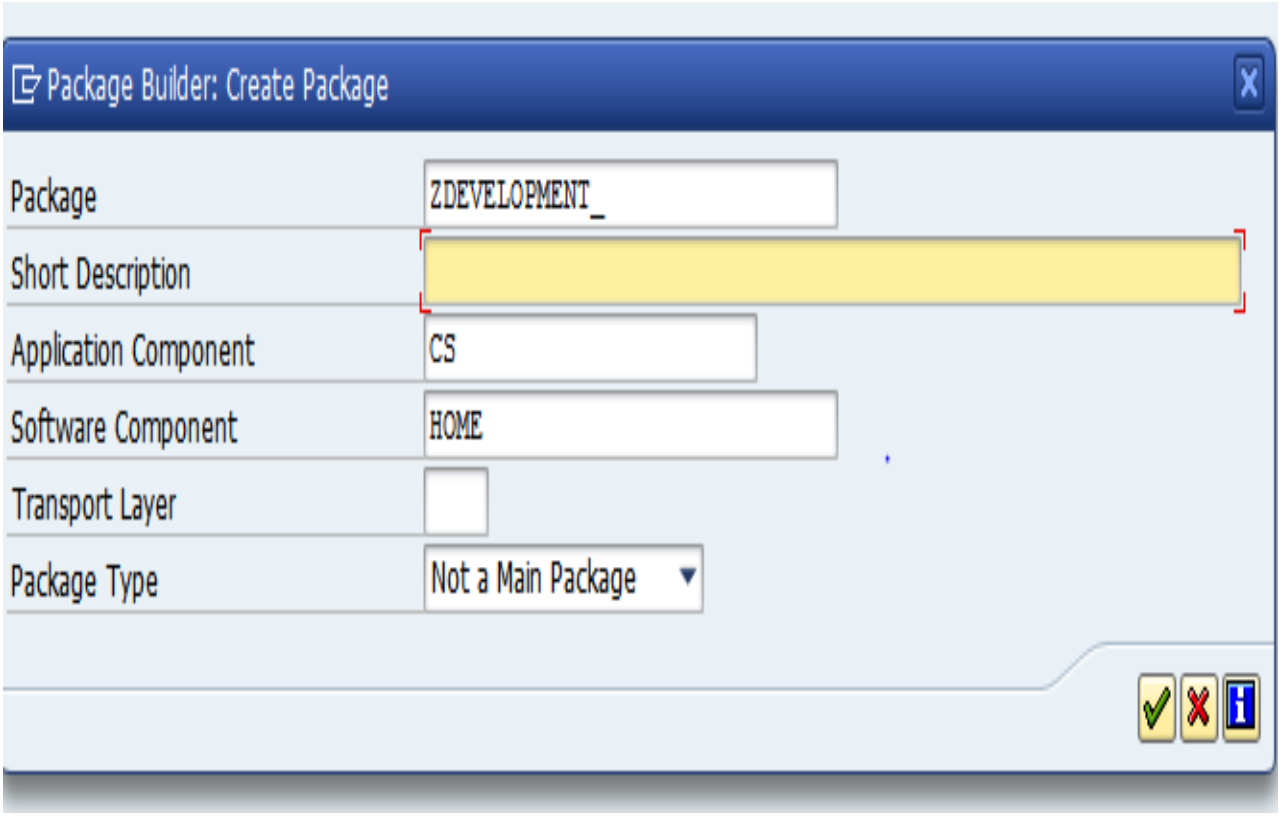
SLIS is a standard package used to find all ALV (ABAP List Viewer) reports.

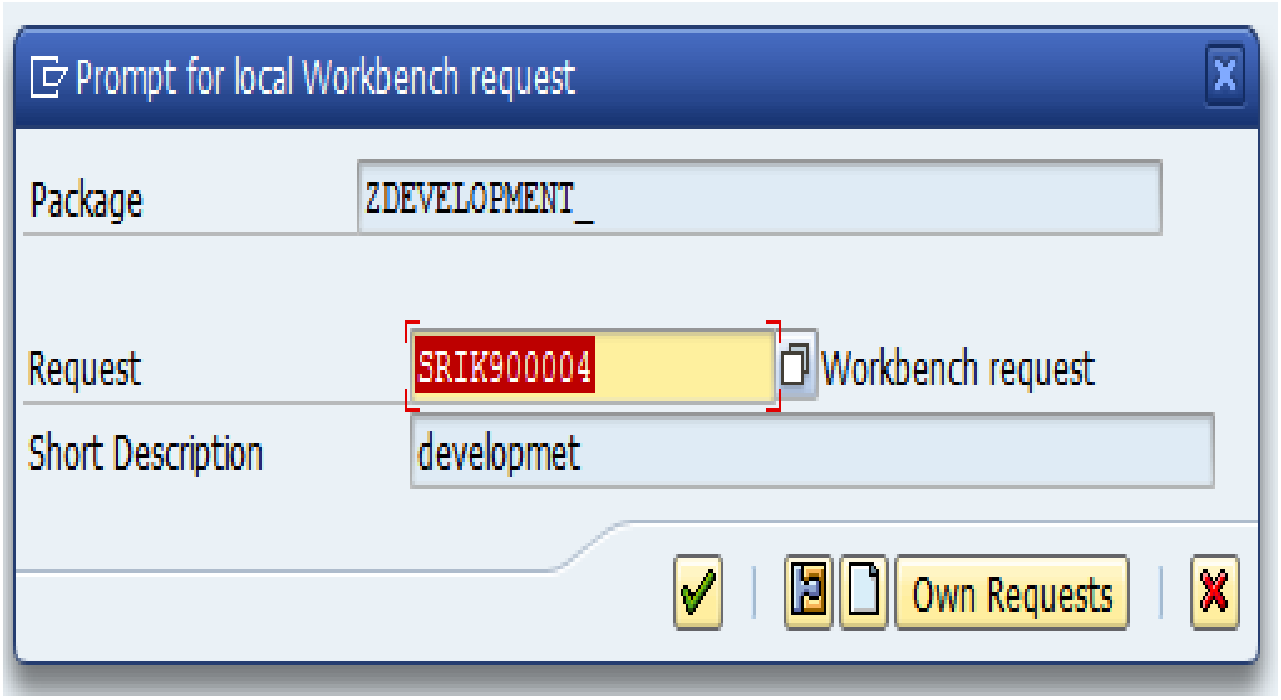
To create a package, do the following steps:

- Go to se21.
- Provide package name i.e., zdevelopment, and click on create.



- Give short description, choose application component as Customer Service and Software component as home. Click on save.





Click on save. In the next window give short description, and click on save.

CHAPTER 6
INTRODUCTION TO ABAP LIST VIEWER

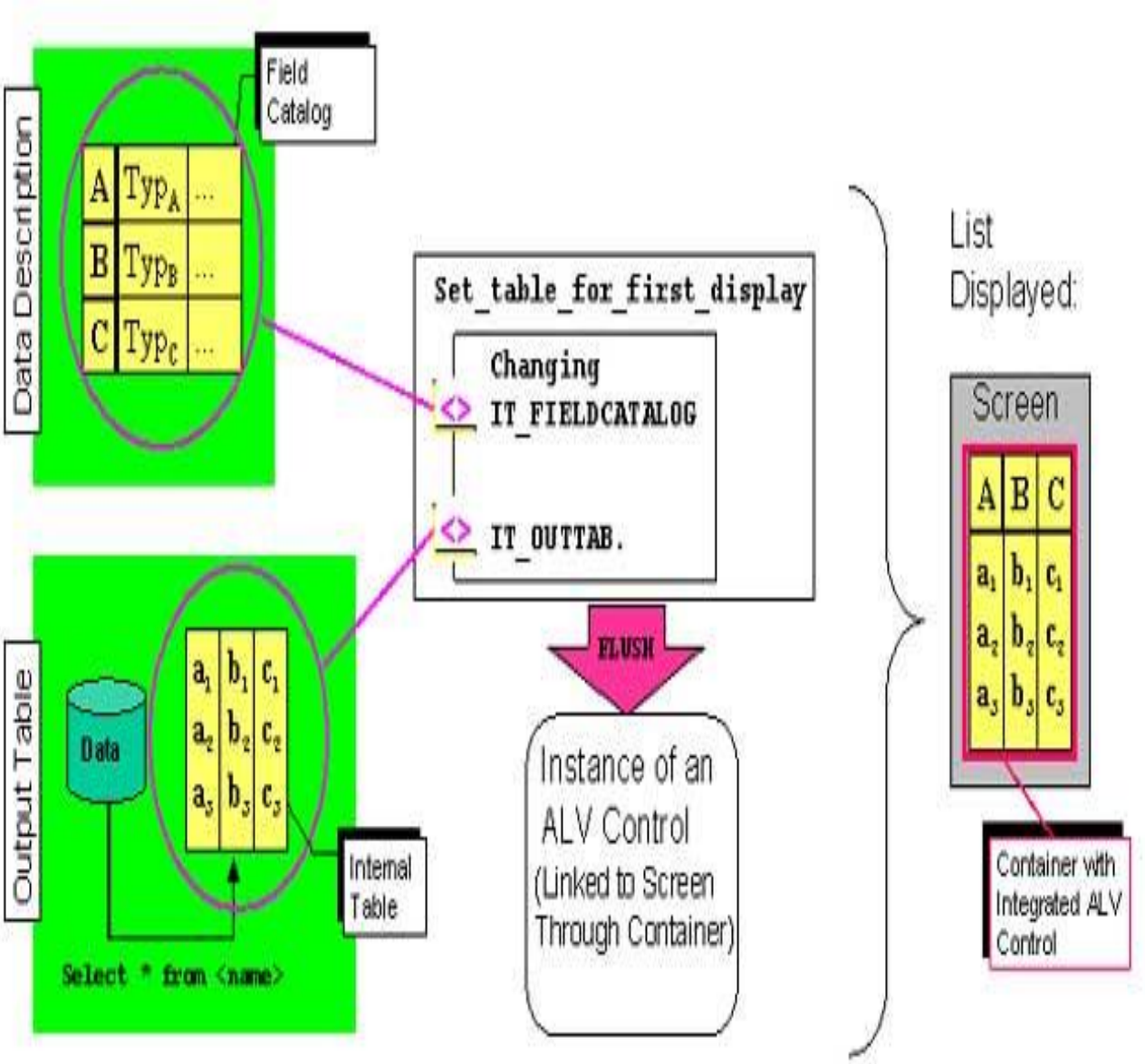
Sap provides a set of ALV (ABAP LIST VIEWER) function modules, which can be put into use to embellish the output of a report. This set of ALV functions is used to enhance the readability and functionality of any report output. Cases arise in sap when the output of a report contains columns extending more than 255 characters in length. In such cases, this set of ALV functions can help choose selected columns and arrange the different columns from a report output and also save different variants for report display. This is a very efficient tool for dynamically sorting and arranging the columns from a report output. The report output can contain up to 90 columns in the display with the wide array of display options.

The commonly used ALV functions used for this purpose are;

1. REUSE_ALV_VARIANT_DEFAULT_GET
2. REUSE_ALV_VARIANT_F4
3. REUSE_ALV_VARIANT_EXISTENCE
4. REUSE_ALV_EVENTS_GET
5. REUSE_ALV_COMMENTARY_WRITE
6. REUSE_ALV_FIELDCATALOG_MERGE
7. REUSE_ALV_LIST_DISPLAY
8. REUSE_ALV_GRID_DISPLAY
9. REUSE_ALV_POPUP_TO_SELECT

The common desired features of any report are "column alignment", sorting, filtering, subtotals, totals etc. To implement these from scratch, a lot of coding effort is to be put. To avoid that we can use a concept called ABAP List Viewer (ALV).

- SIMPLE REPORT
- BLOCK REPORT
- HIERARCHICAL REPORTS



SIMPLE REPORT

Important function modules in these report are -

- Reuse_alv_fieldcatalog_merge
- Reuse_alv_list_display
- Reuse_alv_events_get
- Reuse_alv_grid_display
- Reuse_alv_commentary_write

REUSE_ALV_FIELDCATALOG_MERGE

This function module is used to populate a field catalog which is essential to display the data in ALV.

If the output data is from a single dictionary table and all the columns are selected, then we need not exclusively create the field catalog. Its enough to mention the table name as a parameter (I_structure_name) in the REUSE_ALV_LIST_DISPLAY. But in other cases we need to create it.

Note : Fieldcatalog can be filled manually also by filling up all the required details into the internal table

Important parameters in are:

1. Export:

- I_program_name : report id
- I_internal_tabname : the internal output table
- I_inclname : include or the report name where all the dynamic forms are handled.

2. Changing

- ct_fieldcat : an internal table with the type SLIS_T_FIELDCAT_ALV which is declared in the type pool SLIS.

REUSE_ALV_LIST_DISPLAY

This is the function module which prints the data.

The important parameters are:

1. Export:

- I_callback_program : report id
- I_bypassing_buffer : 'X'
- I_buffer_active : ' '
- I_callback_pf_status_set : routine where a user can set his own pf status or change the functionality of the existing pf status.
- I_callback_user_command : routine where the function codes are handled.

- I_structure name : name of the dictionary table
- Is_Layout : structure to set the layout of the report
- It_fieldcat : internal table with the list of all fields and their attributes which are to be printed (this table can be populated automatically by the function)
- It_events : internal table with a list of all possible events of ALV and their corresponding form names.

2. Tables:

- a. t_outtab : internal table with the data to be output

REUSE_ALV_EVENTS_GET:

Returns table of possible events for a a list type

1. Import:

Et_Events : The event table is returned with all possible CALLBACK events for the specified list type (column 'NAME'). For events to be processed by the Callback, their 'FORM' field must be filled. If the field is initialized, the event is ignored. The entry can be read from the event table, the field 'FORM' filled and the entry modified using constants from the type pool SLIS.

REUSE_ALV_COMMENTARY_WRITE

This is used in the Top-of-page event to print the headings and other comments for the list.

Important Parameters

- It_list_commentary : Internal table with the headings of the type slis_t_listheader.

This internal table has three fields:

1. Typ : 'H' - header, 'S' - selection, 'A' - action
2. Key : only when typ is 'S'.
3. Info : the text to be printed

BLOCK REPORT

This looks like a simple report but this report has the features of sorting and filtering only.This report is used if you have to display more than one report on the output. Technically speaking if you have multiple internal table with data to be displayed as separate blocks then we go for block.

The important functions used for creating this report are:

- REUSE_ALV_BLOCK_LIST_INIT
- REUSE_ALV_BLOCK_LIST_APPEND
- REUSE_ALV_BLOCK_LIST_DISPLAY

REUSE_ALV_BLOCK_LIST_INIT

This function module is used to set the default gui status etc. The parameters are similar to the one used in reuse_alv_list_display or reuse_alv_grid_display

REUSE_ALV_BLOCK_LIST_APPEND

This function module adds the data to the block.

Important Parameters

1.Export :

- is_layout : layout settings for block
- it_fieldcat : field catalog
- I_tabname : internal table name with all all possible events

2.Tables :

- t_outtab : internal table with output data.

REUSE_ALV_BLOCK_LIST_DISPLAY

This function module display the list with data appended by the above function.

Parameters : All the parameters are optional.

Hierarchical Reports

Hierarchical display is used for displaying data that are related. Like sales order and item details. Here sales order details can be the header data whereas them items in the sales order can be the item data

The function module used for this is

REUSE_ALV_HIERSEQ_LIST_DISPLAY

Export:

- I_CALLBACK_PROGRAM
- I_CALLBACK_PF_STATUS_SET
- I_CALLBACK_USER_COMMAND
- IS_LAYOUT
- It_fieldcat
- It_events

- I_tabname_header : Name of the internal table in the program containing the output data of the highest hierarchy level.
- I_tabname_item : Name of the internal table in the program containing the output data of the lowest hierarchy level.
- Is_keyinfo : This structure contains the header and item table field names which link the two tables (shared key).

Tables

- t_outtab_header : Header table with data to be output
- t_outtab_item : Name of the internal table in the program containing the output data of the lowest hierarchy level.

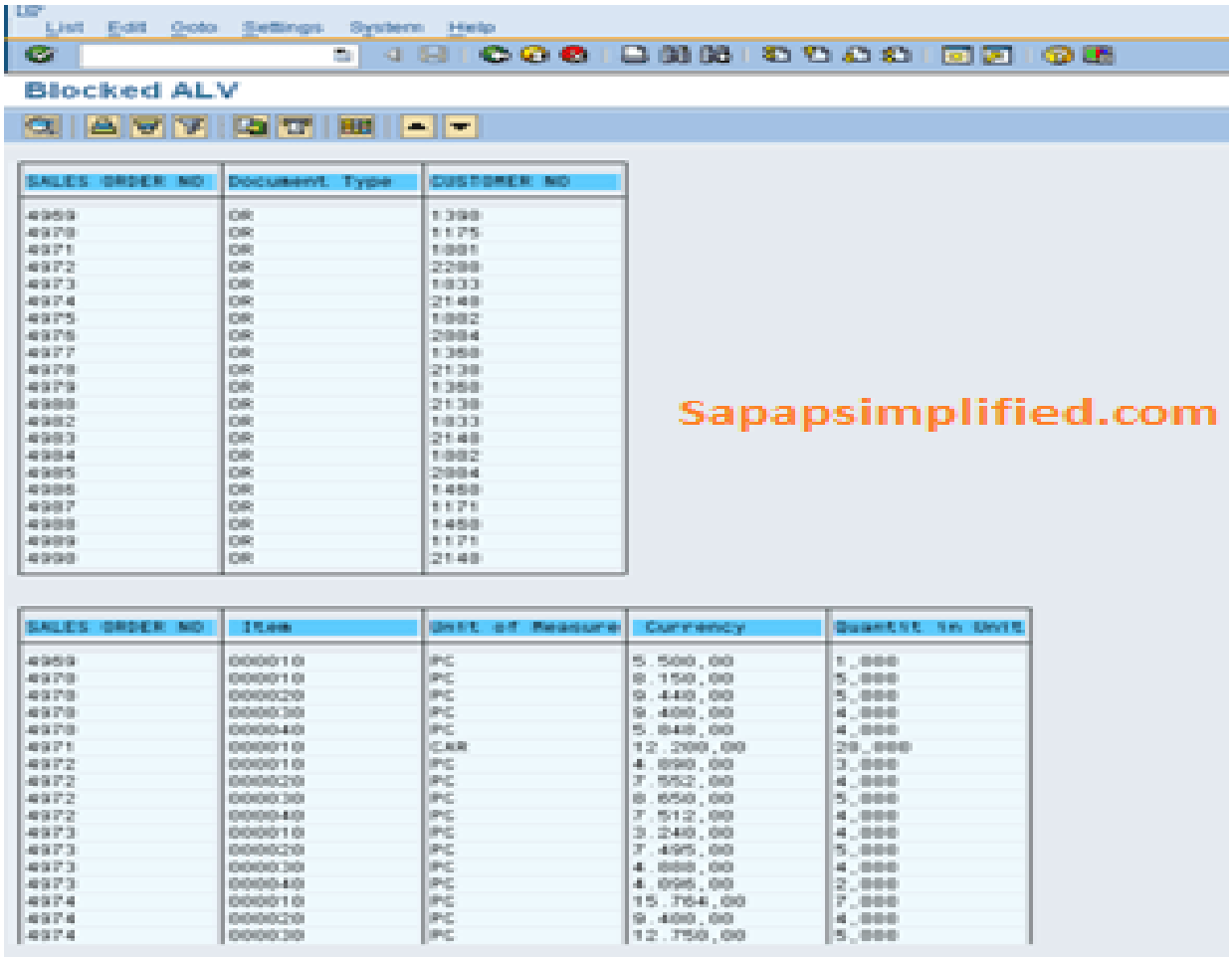
All the definitions of internal tables, structures and constants are declared in a type-pool called **SLIS.** This internal table can be populated automatically by using REUSE_ALV_FIELDCATALOG_MERGE'.

SLIS – this is used for type generic list module.

SDYDO- this is used for dynamic documents.

CHAPTER 7 BLOCKED ALV

Blocked ALV is used to display the output in block wise (display multiple ALV lists in the same screen) a shown below.



Steps to work with blocked ALV:

1. Initialize the blocked ALV by using "**REUSE_ALV_BLOCK_LIST_INIT**" the input for above function module is current program (sy-repid)

2. Append each data internal table to the blocked ALV by using "**REUSE_ALV_BLOCKED_LIST_APPEND** function module

The input for the above function module is

1. Data internal table

2. Field catalog internal table

3. Event internal table

4. Layout internal table

Repeat the step 2 from each block (data internal table)

3. Display the output by using "**REUSE_ALV_BLOCKED_LIST_DISPLAY**" function module

7.2 PROBLEM IDENTIFICATION:

- This study was to initiate an innovative idea in the field of application development to increase the accuracy in the Human Resources field, by using the principles of SAP (System Application Product) using ABAP module preprogramming (User interface). Using blocked alv (abap list viewer). To retrieve valid vendor report.

7.3 USE OF BLOCKED ALV

ALV block list can be used if you want to display multiple ALV lists in the

same screen.

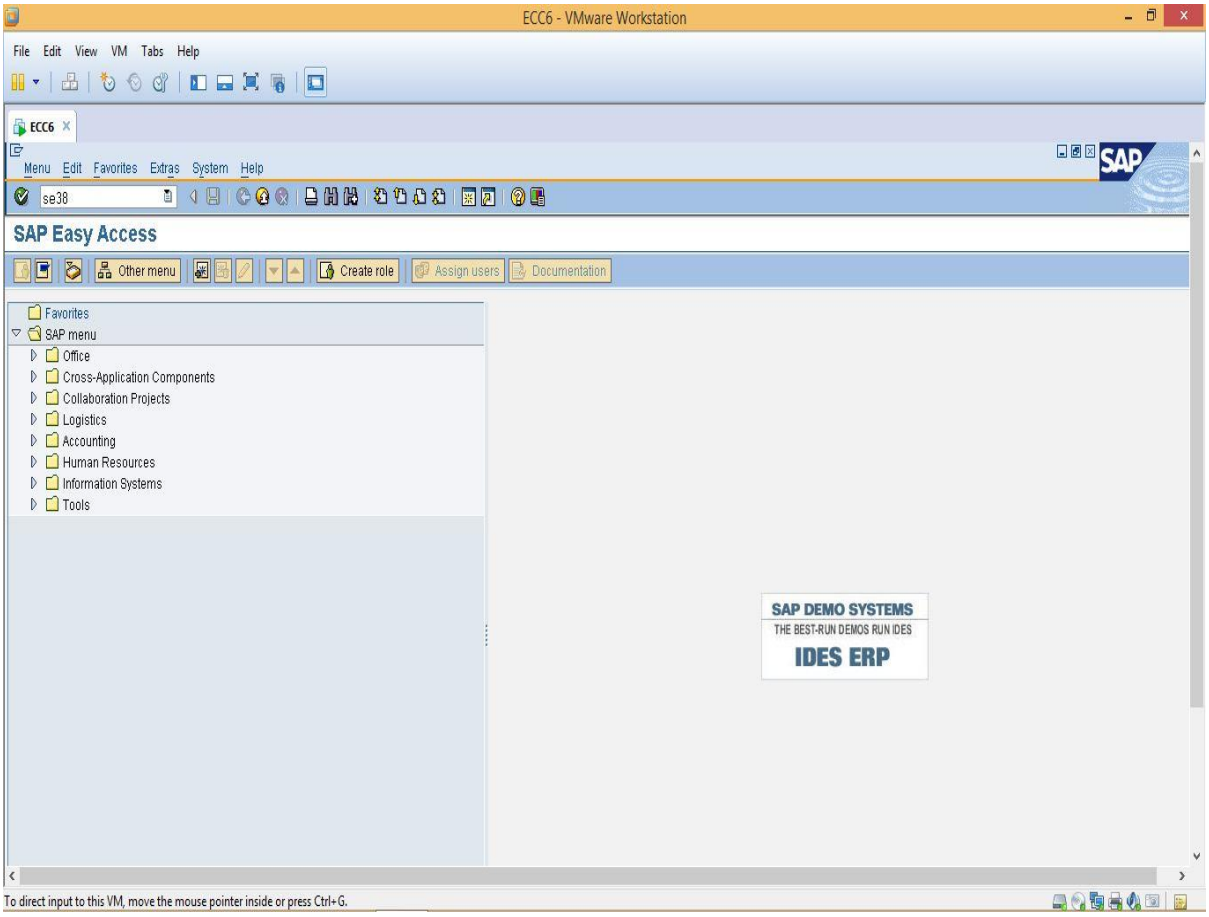
• Blocked ALV will display the output in block wise and user

friendly (filtering,sorting,e-mail etc)

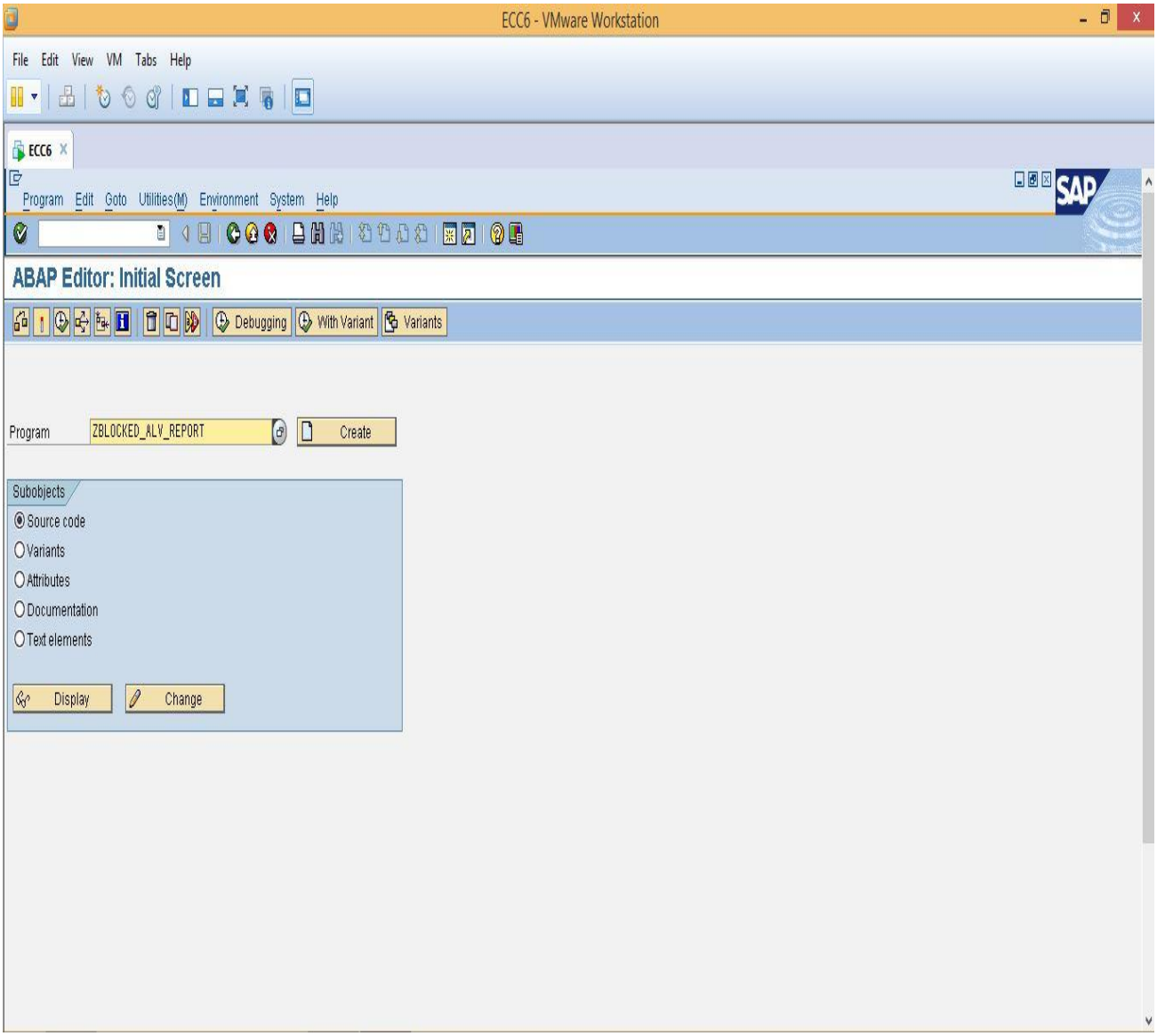
• width of more than 256 characters possible.

7.4 PROGRAM EXECUTION

Open se38

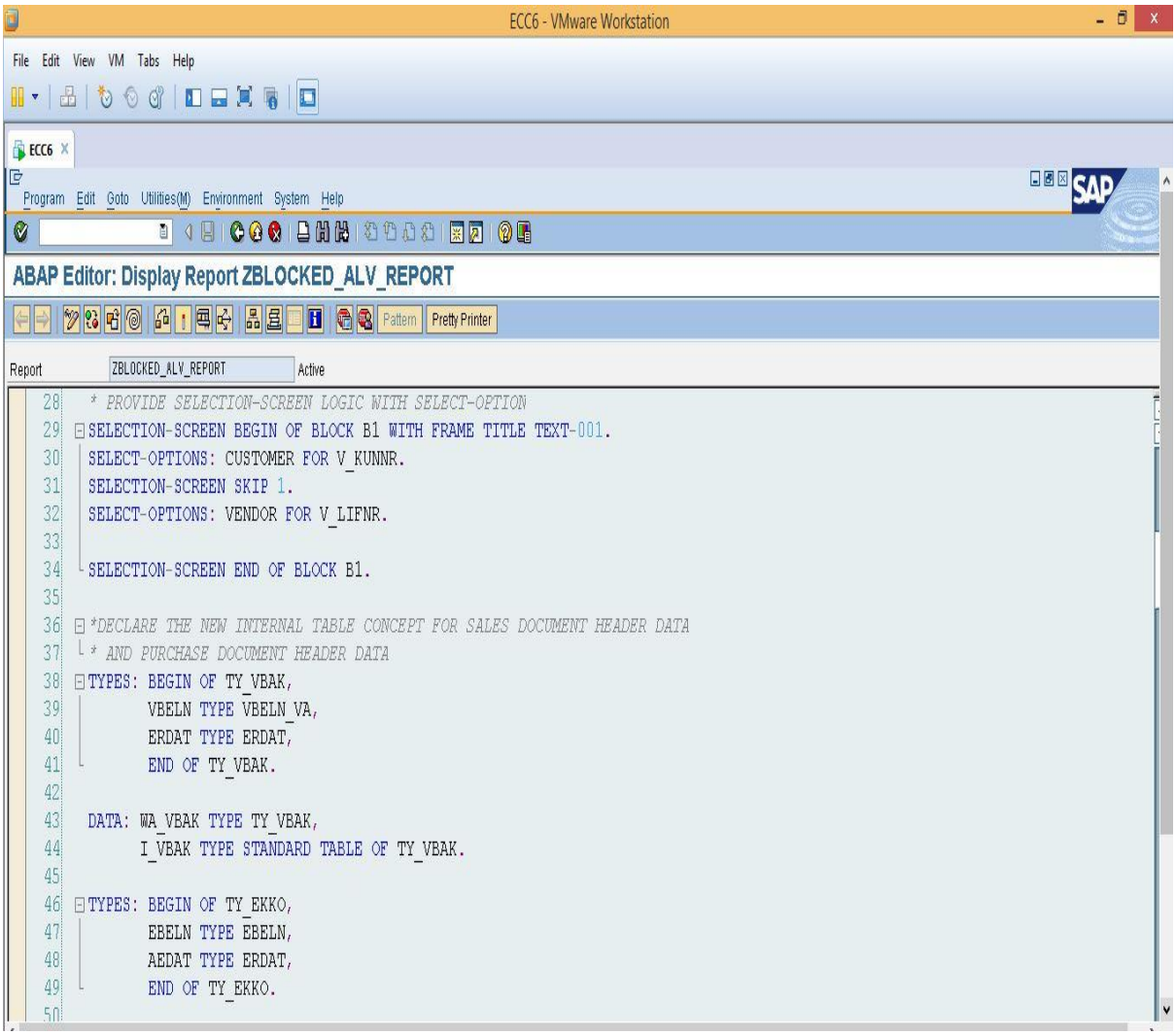


Create program name ZBlocked_alv_report

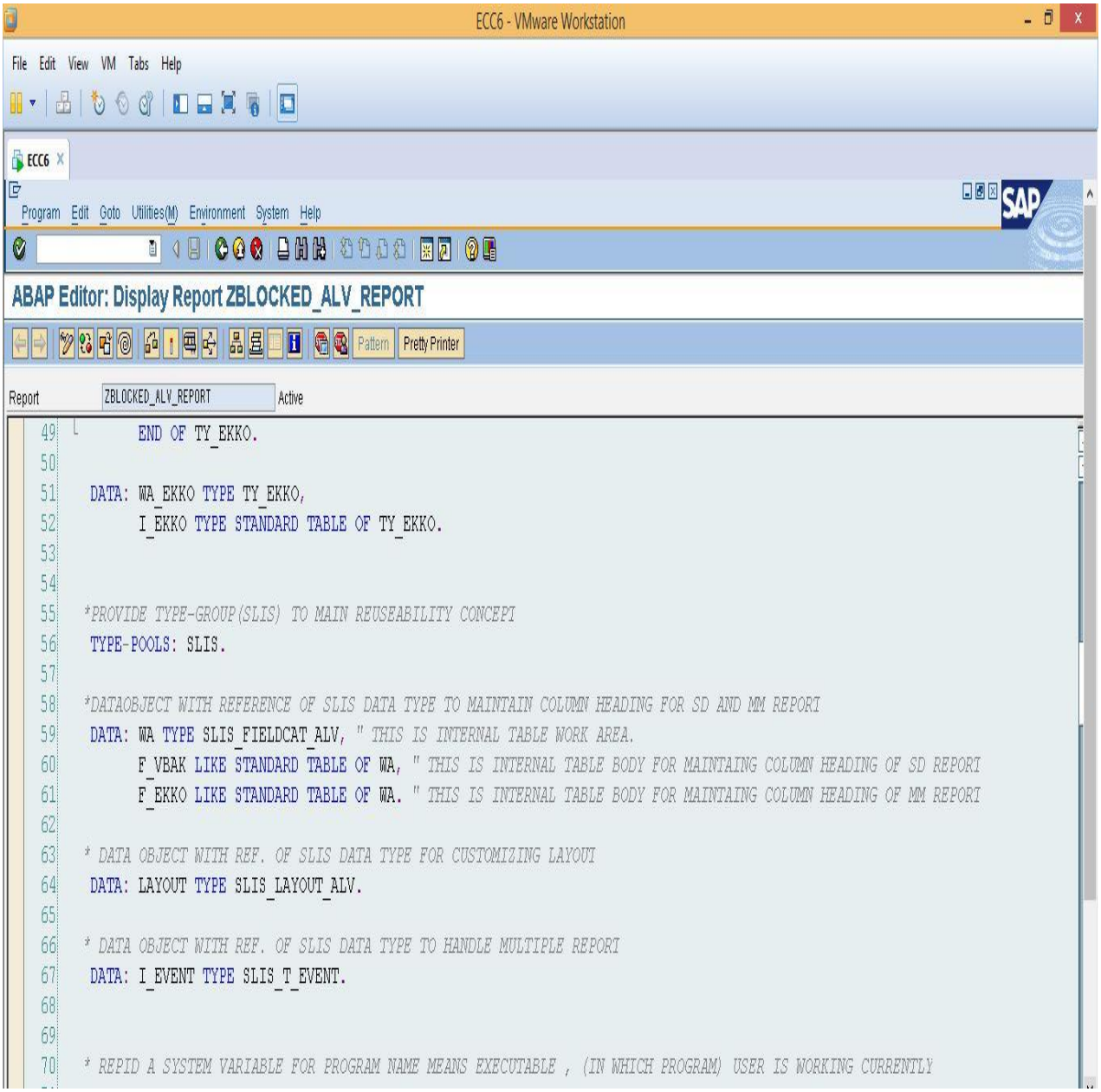


Select the option source code and click on display. If you have already created the program name then we need to click on change for displaying of our code.

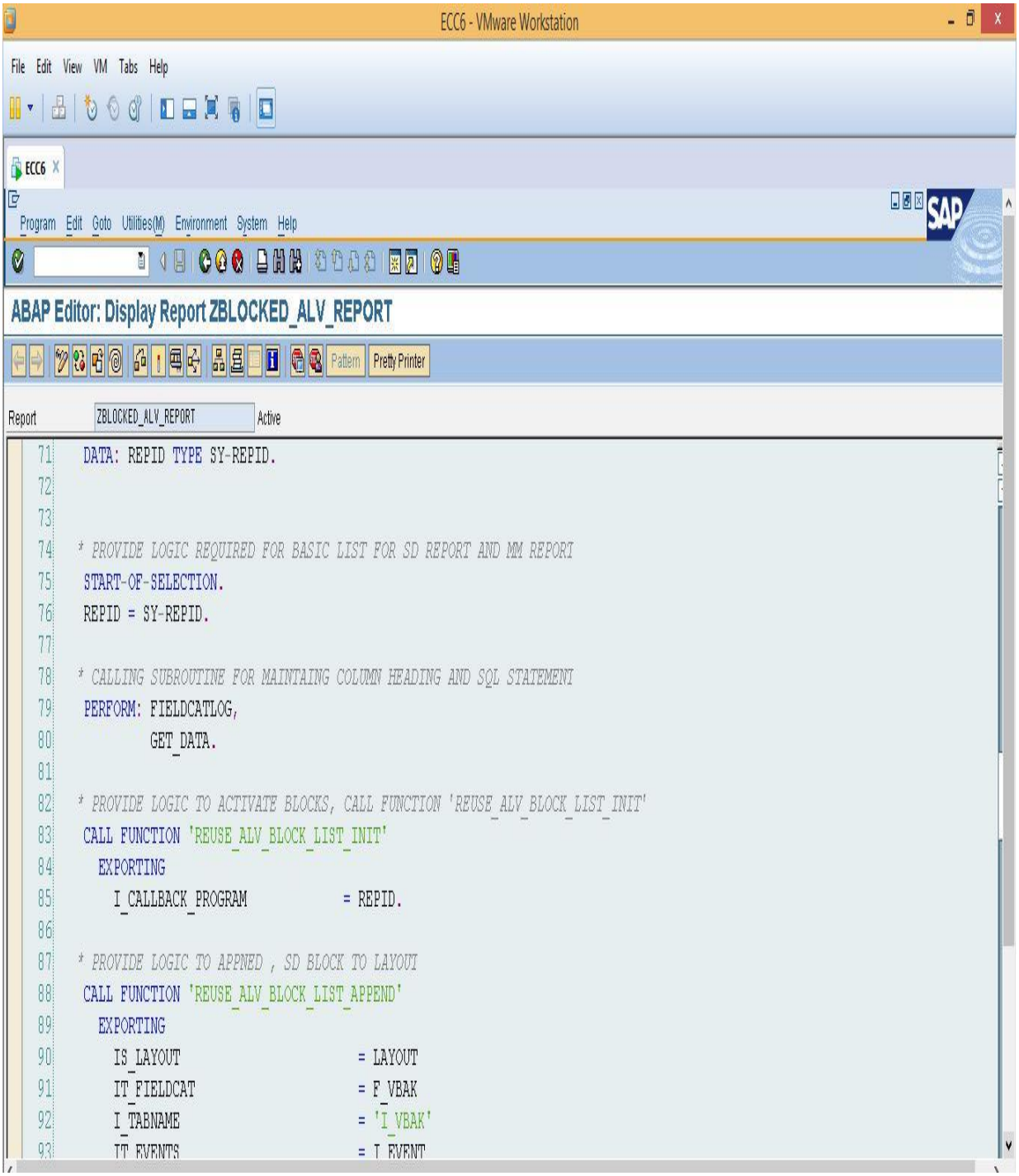
Start program



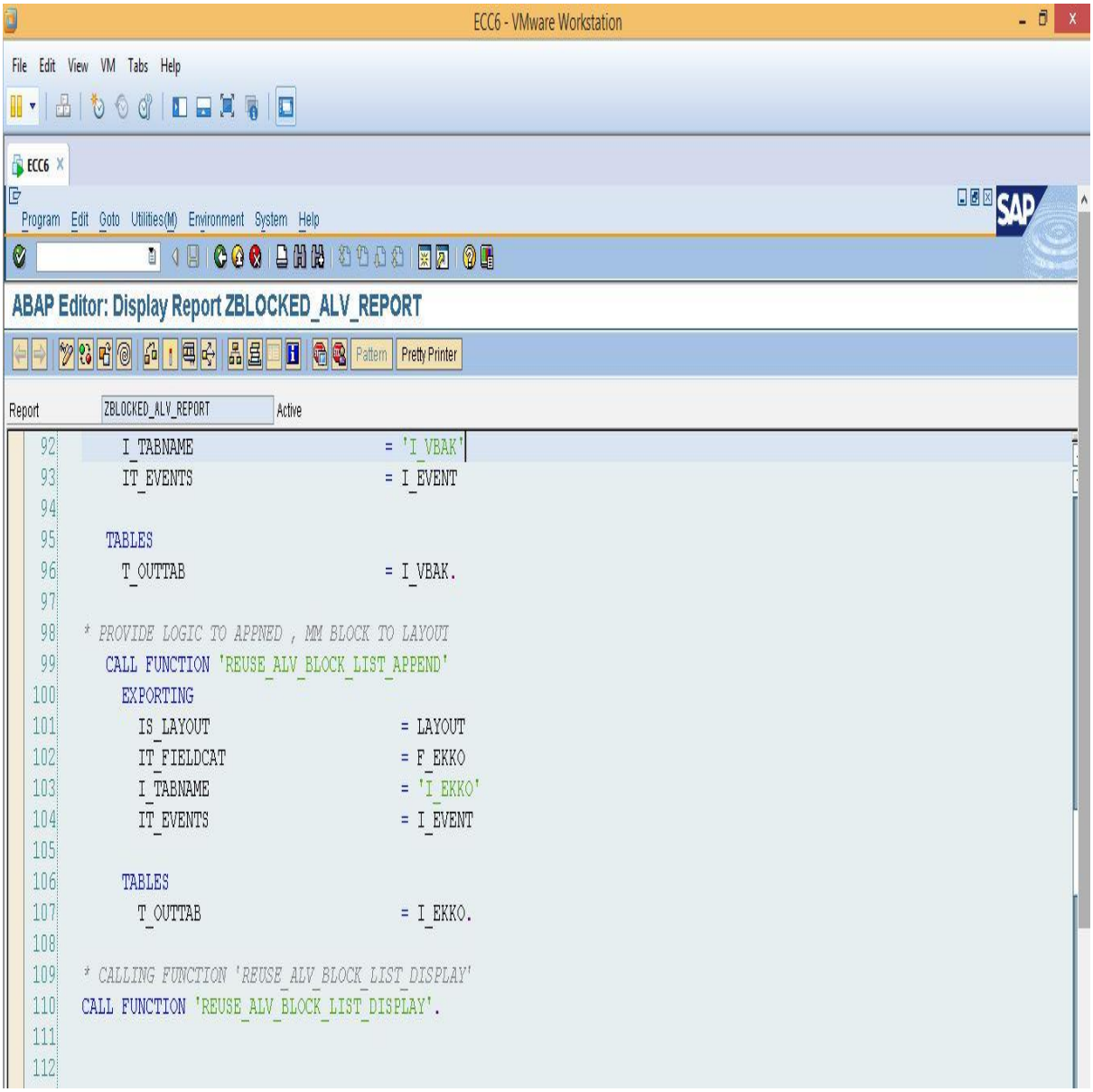
Here we are providing selection-screen logic with select-option. And we are declaring the new internal table concept for sales document header data and purchase document header data.



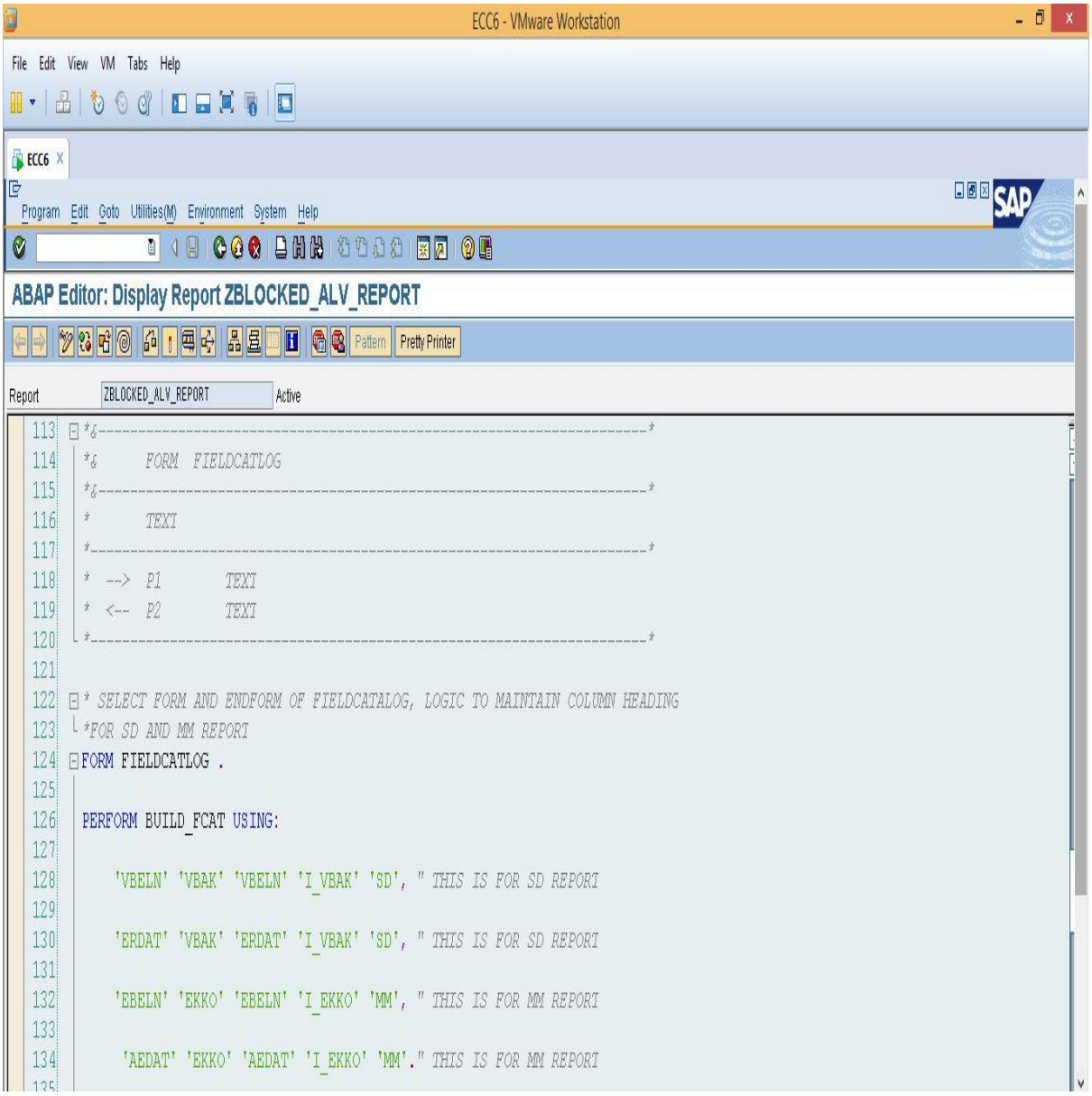
Provide type-group to main reusability concept by type-pools



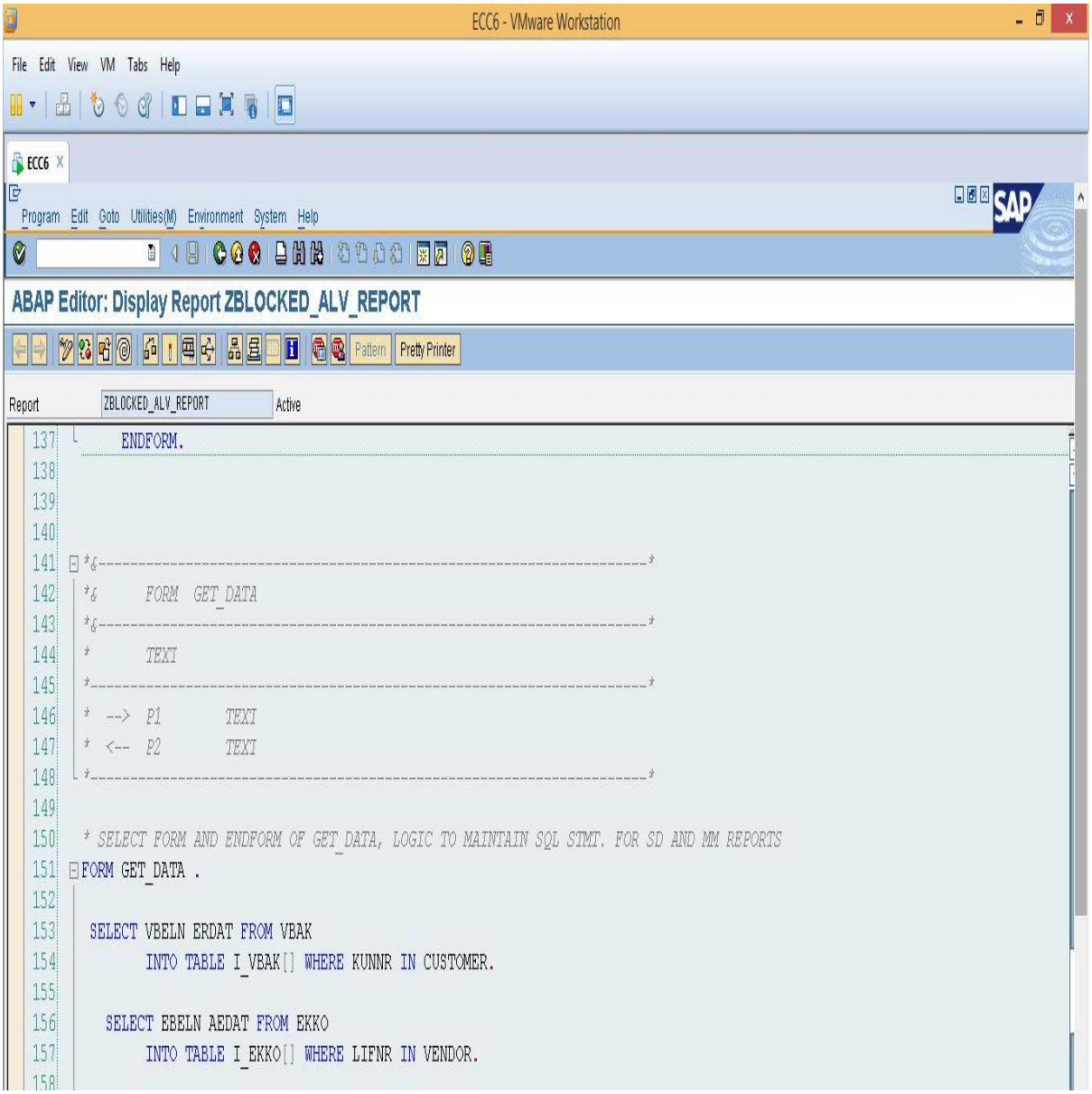
- Provide logic required for basic list for SD report and MM report.
- Calling subroutine for maintain column heading and sql statement.
- Provide logic to activate blocks , call function “reuse_alv_block_list_init”
- Provide logic to append, sd block to layout.



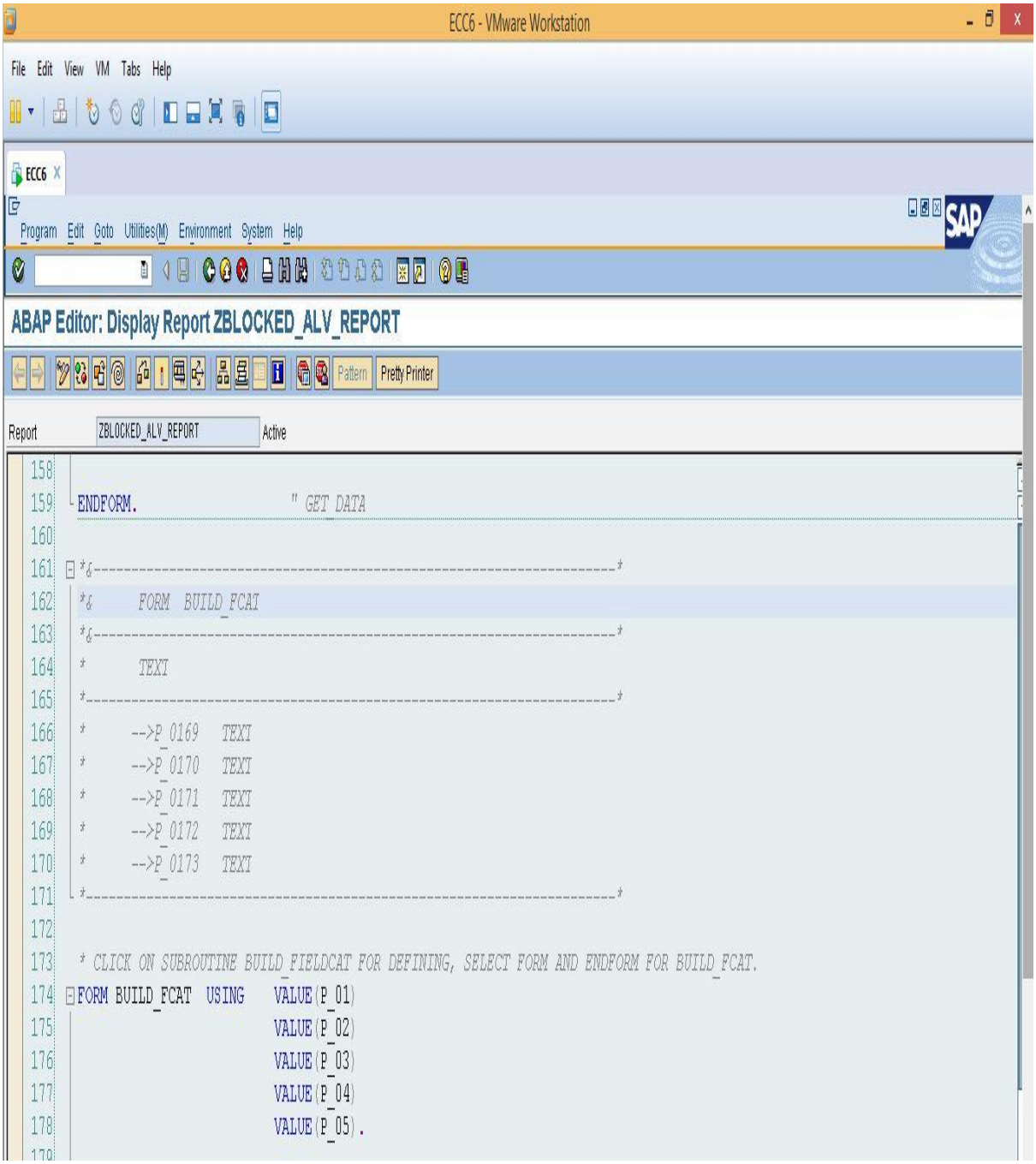
- Provide logic to append ,mm block to layout.
- Calling function “reuse_alv_block_list_append.

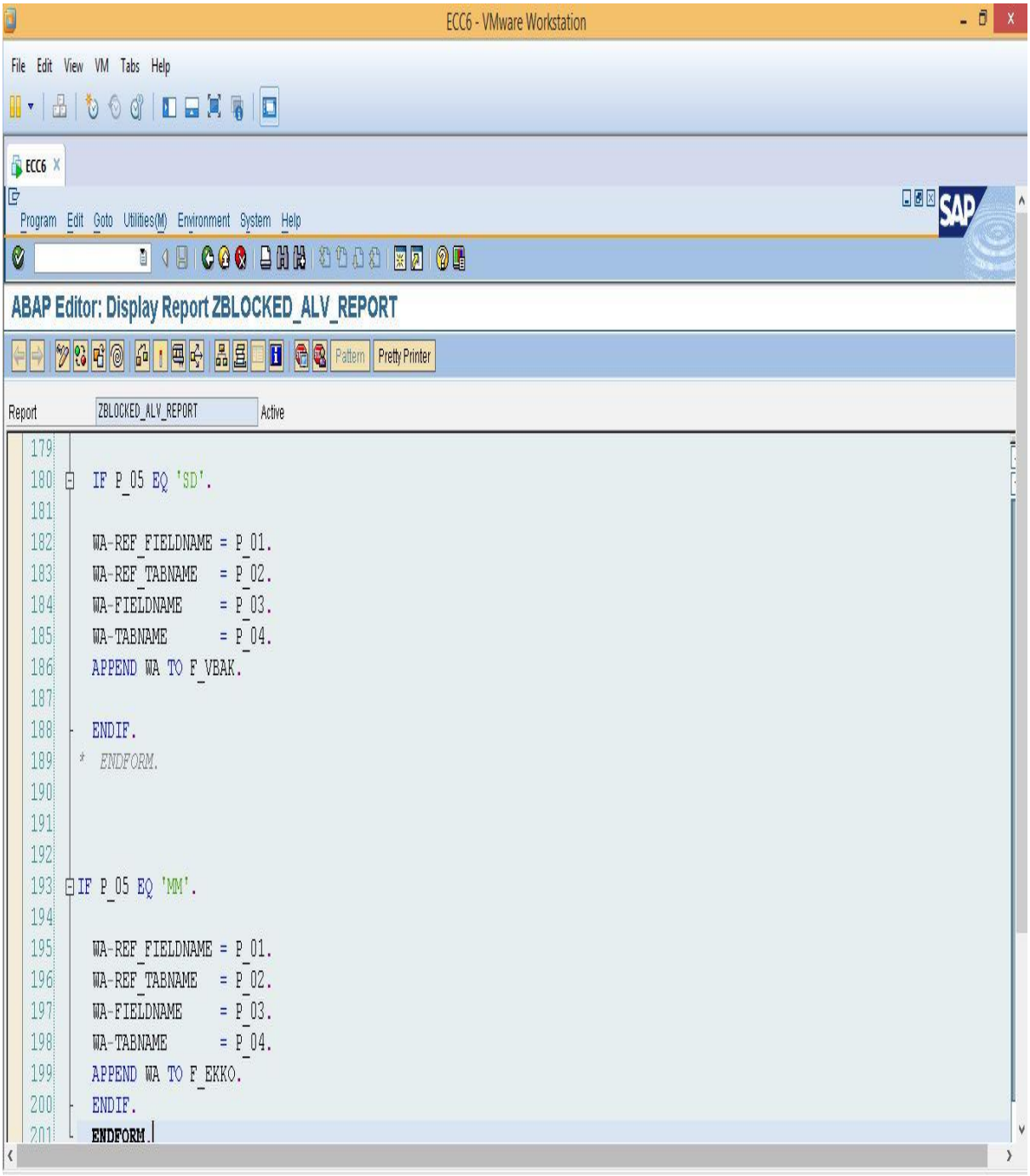


- Select form and endform of fieldcatalog ,logic to maintain column heading.
- For sd and mm report.

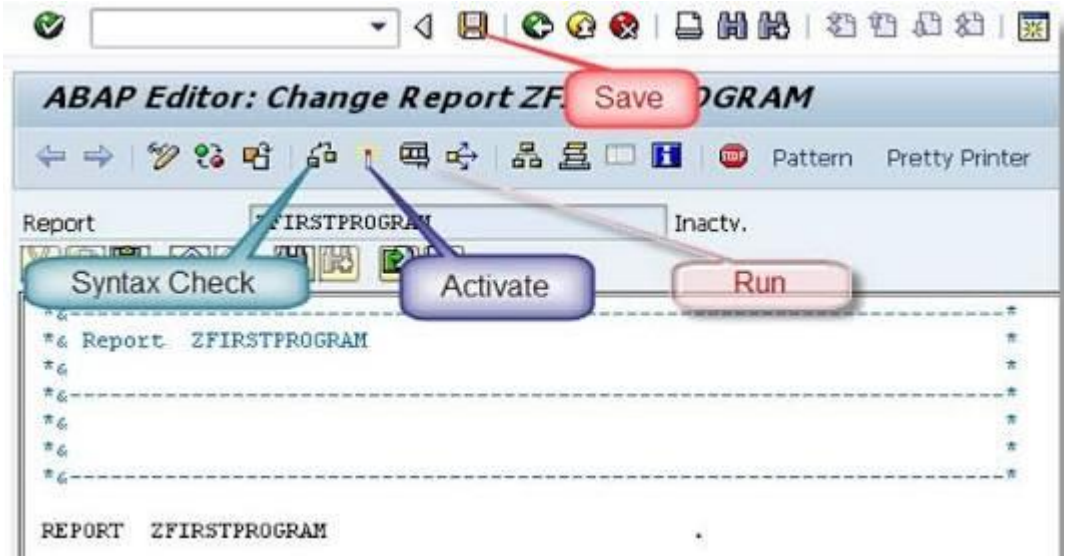


- Select form and endform of get_data, logic to maintain sql stmt. For sd and mm reports

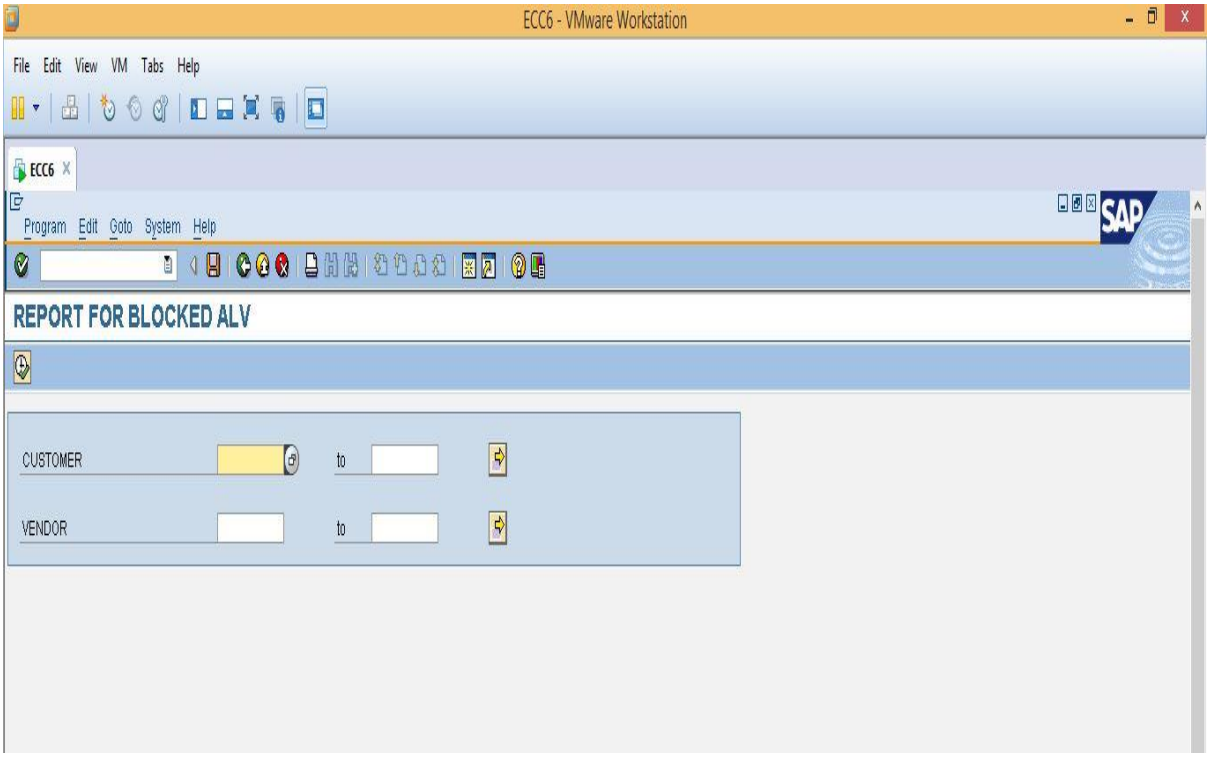

ECC6 - VMware Workstation
File Edit View VM Tabs Help
ECC6
Program Edit Goto Utilities(M) Environment System Help
ABAP Editor: Display Report ZBLOCKED_ALV_REPORT
Pattern
Pretty Printer
Report
ZBLOCKED_ALV_REPORT
Active
158
159 ENDFORM. " GET_DATA
160
161 *&---*
162 *& FORM BUILD_FCAT
163 *&---*
164 * TEXT
165 *---*
166 * -->P_0169 TEXT
167 * -->P_0170 TEXT
168 * -->P_0171 TEXT
169 * -->P_0172 TEXT
170 * -->P_0173 TEXT
171 *---*
172
173 * CLICK ON SUBROUTINE BUILD_FIELDCAT FOR DEFINING, SELECT FORM AND ENDFORM FOR BUILD_FCAT.
174 FORM BUILD_FCAT USING VALUE(P_01)
175 VALUE(P_02)
176 VALUE(P_03)
177 VALUE(P_04)
178 VALUE(P_05).

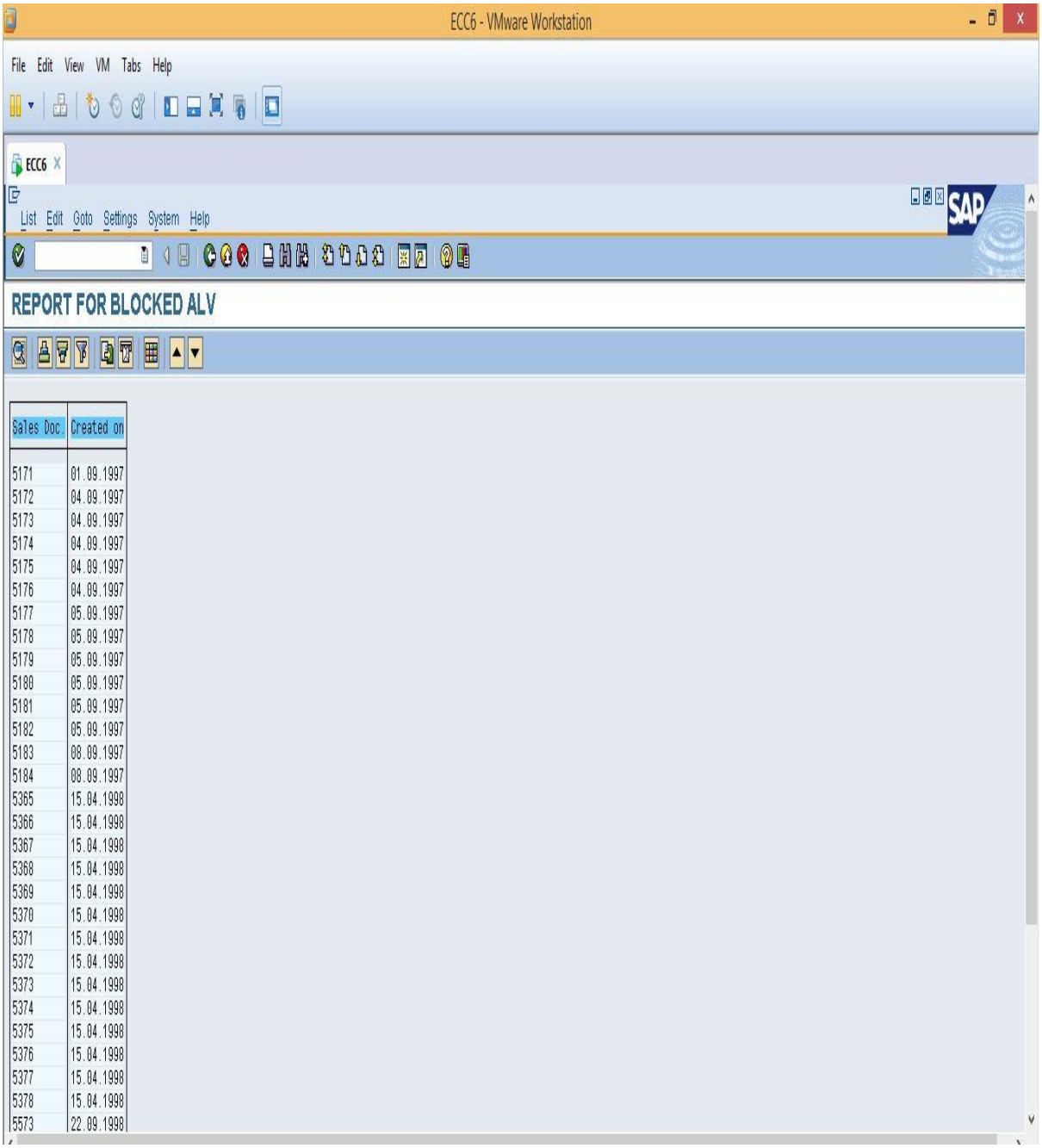

ECC6 - VMware Workstation
File Edit View VM Tabs Help
ECC6
Program Edit Goto Utilities(M) Environment System Help
ABAP Editor: Display Report ZBLOCKED_ALV_REPORT
Pattern
Pretty Printer
Report
ZBLOCKED_ALV_REPORT
Active
179
180 IF P_05 EQ 'SD'.
181
182 WA-REF_FIELDNAME = P_01.
183 WA-REF_TABNAME = P_02.
184 WA-FIELDNAME = P_03.
185 WA-TABNAME = P_04.
186 APPEND WA TO F_VBAK.
187
188 ENDIF.
189 * ENDFORM.
190
191
192
193 IF P_05 EQ 'MM'.
194
195 WA-REF_FIELDNAME = P_01.
196 WA-REF_TABNAME = P_02.
197 WA-FIELDNAME = P_03.
198 WA-TABNAME = P_04.
199 APPEND WA TO F_EKKO.
200 ENDIF.
201 ENDFORM.

Check,Activate and Excute

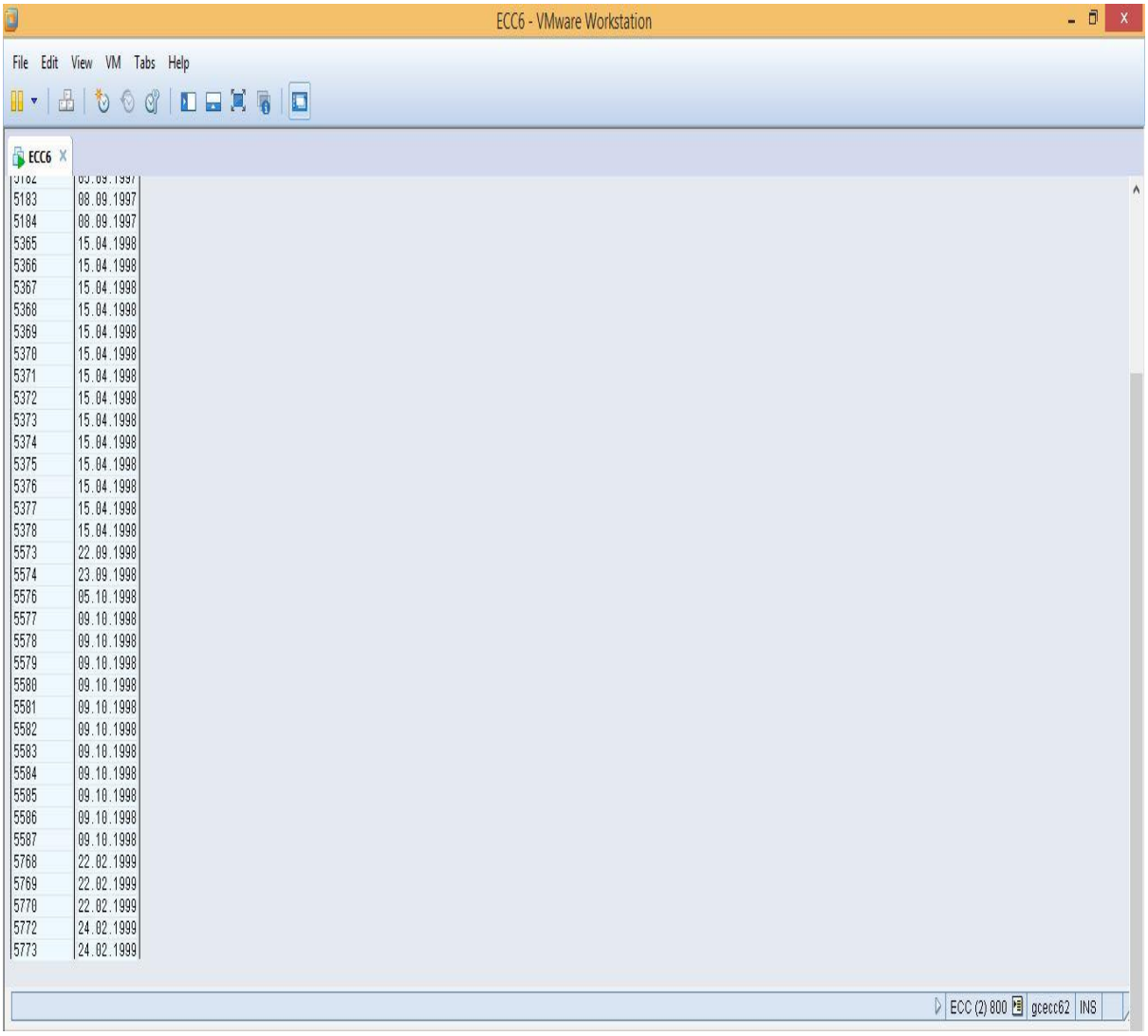


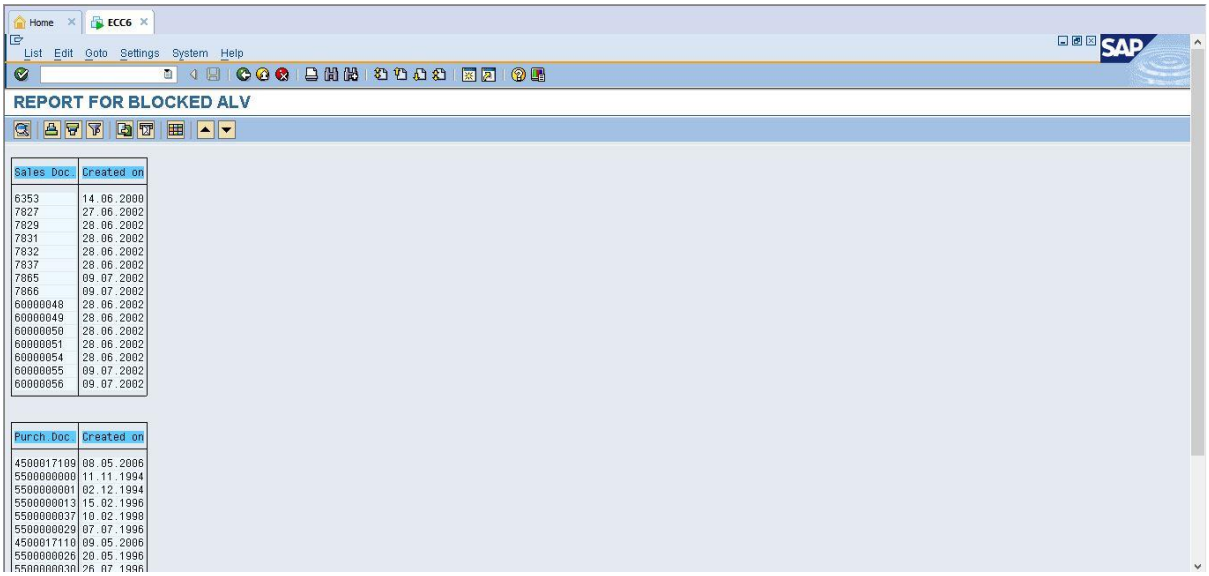
Output:





Sales Doc.	Created on
5171	01.09.1997
5172	04.09.1997
5173	04.09.1997
5174	04.09.1997
5175	04.09.1997
5176	04.09.1997
5177	05.09.1997
5178	05.09.1997
5179	05.09.1997
5180	05.09.1997
5181	05.09.1997
5182	05.09.1997
5183	08.09.1997
5184	08.09.1997
5365	15.04.1998
5366	15.04.1998
5367	15.04.1998
5368	15.04.1998
5369	15.04.1998
5370	15.04.1998
5371	15.04.1998
5372	15.04.1998
5373	15.04.1998
5374	15.04.1998
5375	15.04.1998
5376	15.04.1998
5377	15.04.1998
5378	15.04.1998
5573	22.09.1998


ECC6 - VMware Workstation
File Edit View VM Tabs Help
ECC6
5183 08.09.1997
5184 08.09.1997
5365 15.04.1998
5366 15.04.1998
5367 15.04.1998
5368 15.04.1998
5369 15.04.1998
5370 15.04.1998
5371 15.04.1998
5372 15.04.1998
5373 15.04.1998
5374 15.04.1998
5375 15.04.1998
5376 15.04.1998
5377 15.04.1998
5378 15.04.1998
5573 22.09.1998
5574 23.09.1998
5576 05.10.1998
5577 09.10.1998
5578 09.10.1998
5579 09.10.1998
5580 09.10.1998
5581 09.10.1998
5582 09.10.1998
5583 09.10.1998
5584 09.10.1998
5585 09.10.1998
5586 09.10.1998
5587 09.10.1998
5768 22.02.1999
5769 22.02.1999
5770 22.02.1999
5772 24.02.1999
5773 24.02.1999
ECC (2) 800
gcecc62
INS



Sales Doc.	Created on
6353	14.06.2000
7827	27.06.2002
7829	28.06.2002
7831	28.06.2002
7832	28.06.2002
7837	28.06.2002
7865	09.07.2002
7866	09.07.2002
60000048	28.06.2002
60000049	28.06.2002
60000050	28.06.2002
60000051	28.06.2002
60000054	28.06.2002
60000055	09.07.2002
60000056	09.07.2002

Purch Doc.	Created on
4500017109	08.05.2006
5500000000	11.11.1994
5500000001	02.12.1994
5500000013	15.02.1996
5500000037	10.02.1998
5500000029	07.07.1996
4500017110	09.05.2006
5500000026	20.05.1996
5500000030	26.07.1996

CHAPTER 8

CONCLUSION

- Sap has multilayered integrated framework.
- To ensure adequate protection, security measures must be factored into all layers of SAP infrastructure.
- With client/server architecture, SAP systems include many constitutes a layer of the SAP security infrastructure.
- Security is often not a priority in an implementation and as a result, the default security is not strong.
- SAP security functionality could be enhanced using various measures as discussed here.
- Enterprises must develop a security strategy to ensure a secure and functional SAP needs continuous monitoring and improvement of its security features.

SAP ERP delivers a comprehensive set of integrated, cross-functional business processes. It's performing well in the following aspects such as

- Controlling and maintaining inventory in MM department.
- Reducing inventory costs.
- Improve alignment of strategies and operations.
- Improving productivity and insight.
- Reducing risk.
- Improves financial management and corporate governance.
- Optimize IT spending and to provide immediate access to enterprise information.

This application can be used to generate vendors report like vendors name and product similarly the product will also have another block consisting of product manufacture date expiry date etc. similarly there might be various block therefore this blocked ALV displays all the blocks on same screen which is useful for accessing report quickly and provides accurate information.

CHAPTER 9

FUTURE SCOPE

- SAP is still investing heavily in the development of ABAP, partly because most of their core software (ECC, CRM etc.) is written in ABAP, but also because it is an excellent and robust platform for developing business applications.

- You can follow the development and announcement of cutting edge features in ABAP by following the blog posts of Horst Keller on SCN: Horst Keller's Stuff | SCN.

- Java gained prominence in SAP due to a number of factors, such as the acquisition of Top Tier in 2001. Also, products like SAP PI (formerly XI) are developed in Java. **SAP ABAP** is one of the fourth generation language which provides immense career opportunity for SAP ABAP programmer. SAP is mostly an ERP system written on ABAP programming language.

- There is huge scope for ABA Per in India. ABAP programmer can build their career with SAP user by providing their services & marketing SAP products deliver proven value & also can make their career as independent developers by writing correct code for programming applications, providing best methodologies, advance software tools and best coding & programming

Reference

The following resources contains information about SAP

https://www.tutorialspoint.com/sap/

The following resources contain additional information on SAP

https://www.saponlinetutorials.com/

Wiki page for SAP- https://en.wikipedia.org/?title=SAP_R/3

Official page of SAP- https://www.sap.com/index.html

Some useful books are:

SAP R/3 for everyone

SAP R/3 Handbook third edition