# Szoftverfejlesztés az NI (Emerson) nagyvállalati környezetében – SQL, PL/SQL, ERP

# SQL

- Structured Query Language
- Standard language to access relational databases
- The most basic syntax of SQL starts with a SELECT clause and a FROM clause. A SELECT clause tells the database WHAT you want to select. A FROM clause tells the database where to find that data.

```
SELECT table_name
  FROM all_tables
ORDER BY table_name;
```

# SQL - DML

- DML are the keywords you use to access and manipulate data
- SELECT, UPDATE, INSERT, MERGE, DELETE, etc

```
INSERT INTO whatever (column1, column2, column3)
  VALUES (1, 'Text Data', sysdate );

SELECT column1, column2, column3
  FROM whatever;

COMMIT;
```

# SQL - DDL

- DDL are the keywords you use to create objects such as views, tables and procedures etc.
- CREATE TABLE, ALTER VIEW, CREATE OR REPLACE PROCEDURE, etc.

```
CREATE TABLE whatever (
    column1 NUMBER NOT NULL,
    column2 VARCHAR2(10),
    column3 DATE );
```

# PL/SQL

- PL/SQL is the Oracle native programming language that provides database-centric application development. It can natively call static SQL and provides multiple methods of calling dynamic SQL
- PL/SQL is a procedural language like C++, Java, ADA, etc. It has variables and variable declarations, conditional controls like IF and CASE. It has looping structures such as LOOP, FOR LOOP and the WHILE LOOP. PL/SQL uses SQL to use, manipulate and save data to the database.

```
BEGIN
  -- A PL/SQL cursor
  FOR c1 IN (
    SELECT * FROM whatever ) -- This is SQL called by PL/SQL
  LOOP

    DBMS_OUTPUT.PUT_LINE( 'Column1 is: ' ||TO_CHAR(c1.column1) ||
                          ', Column2 is: ' || c1.column2 ||
                          ', Column3 is: ' || TO_CHAR(c1.column3 ) );
  END LOOP;

END;
```

# Differences between SQL and PL/SQL

- SQL is a data-oriented language for selecting and manipulating sets of data. PL/SQL is a procedural language to create applications. You don't normally have a "SQL application". You normally have an application that uses SQL and a relational database on the back-end. PL/SQL can be the application language just like Java or PHP can. SQL may be the source of data for your screens, web pages and reports. PL/SQL might be the language you use to build, format and display those screens, web pages and reports.

# PL/SQL - Declarations

- Variables are placeholders that store the values that can change through the PL/SQL Block.
- Local variables - These are declared in a inner block and cannot be referenced by outside Blocks.
- Global variables - These are declared in a outer block and can be referenced by its itself and by its inner blocks.

```
DECLARE
  var_num1 number;
  var_num2 number;
BEGIN
  var_num1 := 100;
  var_num2 := 200;
  DECLARE
    var_mult number;
    BEGIN
       var_mult := var_num1
* var_num2;
  END;
END;
/
```

# PL/SQL - Statements

IF THEN ELSE STATEMENT

1)

```
IF condition
THEN
 statement 1;
ELSE
 statement 2;
END IF;
```

2)

```
IF condition 1
THEN
 statement 1;
 statement 2;
ELSIF condtion2 THEN
 statement 3;
ELSE
 statement 4;
END IF;
```

# PL/SQL - Statements

- Iterative Statements are used when we want to repeat the execution of one or more statements for specified number of times.
- There are three types of loops in PL/SQL:

  - Simple Loop
    - LOOP
    - statements;
    - EXIT;
    - {or EXIT WHEN condition;}
    - END LOOP;

# PL/SQL - Statements

- Iterative Statements are used when we want to repeat the execution of one or more statements for specified number of times.
- There are three types of loops in PL/SQL:

  - While Loop
    - WHILE <condition>  LOOP
    -  statements;
    - END LOOP;

# PL/SQL - Statements

- Iterative Statements are used when we want to repeat the execution of one or more statements for specified number of times.
- There are three types of loops in PL/SQL:

  - For Loop
    - FOR counter IN val1..val2
    - LOOP statements;
    - END LOOP;
    - val1 - Start integer value.
    - val2 - End integer value.

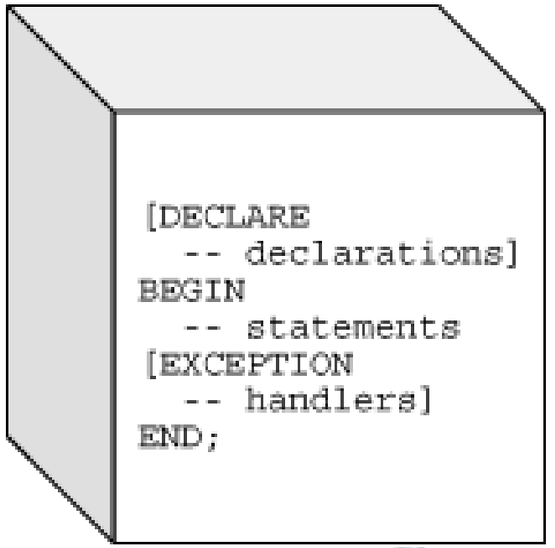
# PL/SQL – Dynamic SQL

- **Dynamic SQL** is a programming methodology for generating and running SQL statements at run time. It is useful when writing general-purpose and flexible programs like ad hoc query systems, when writing programs that must run database definition language (DDL) statements, or when you do not know at compilation time the full text of a SQL statement or the number or data types of its input and output variables.
  - For example, a SELECT statement that includes an identifier that is unknown at compile time (such as a table name) or a WHERE clause in which the number of subclauses is unknown at compile time

- The **EXECUTE IMMEDIATE** statement builds and runs a dynamic SQL statement in a single operation.

# PL/SQL – Blocks

- A PL/SQL block has three basic parts:
  - a declarative part (**DECLARE**),
  - an executable part (**BEGIN .. END**),
  - and an exception-handling (**EXCEPTION**) part that handles error conditions.
- Only the executable part is required. The optional declarative part is written first, where you define types, variables, and similar items. These items are manipulated in the executable part. Exceptions raised during execution can be dealt with in the exception-handling part.



# PL/SQL – Procedure

- A **subprogram** is a program unit/module that performs a particular task. These subprograms are combined to form larger programs. This is basically called the 'Modular design'. A subprogram can be invoked by another subprogram or program which is called the calling program.
- A subprogram can be created:
  - At schema level
  - Inside a package
  - Inside a PL/SQL block

```
CREATE OR REPLACE PROCEDURE greetings AS
BEGIN
   dbms_output.put_line('Hello World!');
END;
/
```

# PL/SQL – Function

- A PL/SQL function is same as a procedure except that it returns a value.

```
CREATE OR REPLACE FUNCTION totalCustomers RETURN number
IS
   total number(2) := 0;
BEGIN
   SELECT count(*) into total
   FROM customers;

   RETURN total;
END;
/
```

# PL/SQL – Package

- PL/SQL lets you bundle logically related types, variables, cursors, and subprograms into a package, a database object that is a step above regular stored procedures. The packages defines a simple, clear, interface to a set of related procedures and types that can be accessed by SQL statements.
- Packages usually have two parts: a specification and a body. The specification defines the application programming interface; it declares the types, constants, variables, exceptions, cursors, and subprograms. The body fills in the SQL queries for cursors and the code for subprograms.

# PL/SQL – Exception Handling

- In PL/SQL, a warning or error condition is called an exception.
- Exceptions can be internally defined (by the run-time system) or user defined. Examples of internally defined exceptions include division by zero and out of memory.
- When an error occurs, an exception is raised. That is, normal execution stops and control transfers to the exception-handling part of your PL/SQL block or subprogram.
  - Internal exceptions are raised implicitly (automatically) by the run-time system.
  - User-defined exceptions must be raised explicitly by RAISE statements, which can also raise predefined exceptions.

- To handle raised exceptions, you write separate routines called exception handlers.
- After an exception handler runs, the current block stops executing and the enclosing block resumes with the next statement.
- If there is no enclosing block, control returns to the host environment.

# Oracle E-Business Suite

# Integrált vállalatirányítási rendszer

# ERP systems

- ERP = Enterprise Resource Planning
- Data of the business processes are collected in one system
- Automates the tasks generated by business processes
- Based on database
- Unified user interface (GUI)
- Built of modules in most cases (e.g. financial, customer, supplier, manufacturing data), generally these can be bought and installed independently from each other
- Customizable
- Appropriate for making integrated analysis, executive summary
- Can be useful for small companies
- Expected for big companies

# ERP systems

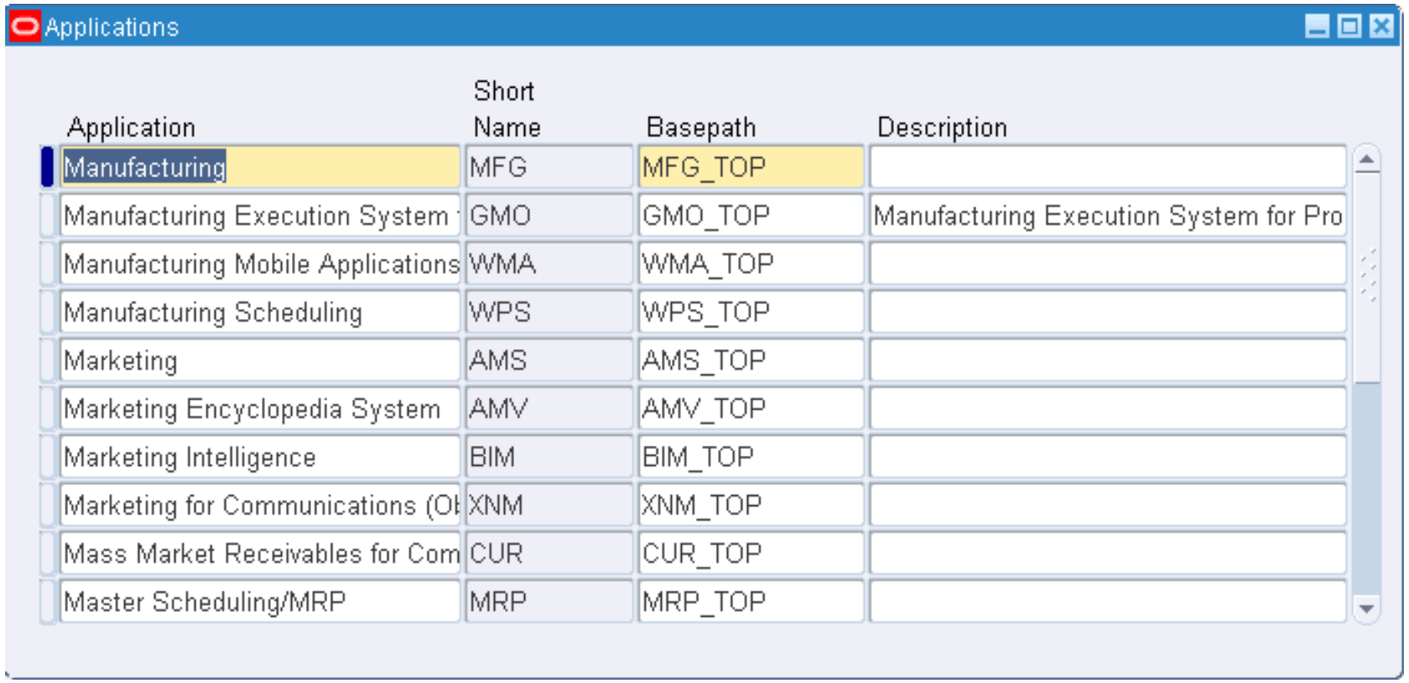
- Well-known ERP systems:
  - Oracle Applications (Oracle)
  - SAP / Small Business One (SAP SBO)
  - AXAPTA (Microsoft)
  - Navision (Microsoft)
  - IFS Applications (Swedish)
  - LIBRA (Hungarian: Volán Elektronika)
  - Microsoft Dynamics
  - etc.

# Modules (applications) in Oracle Applications

- Oracle Applications is an ERP system made up of modules that can be purchased and installed independently from each other
- These applications cover the business processes of the given business areas
- Modules can be divided into 2 groups:
  - **ERP**: Financials, Manufacturing, Order Management, Purchasing, Supply Chain Planning, HRMS (human resource management)
  - **CRM**: Business Intelligence, Marketing, Sales, Call Center

# Modules (applications) in Oracle Applications

- Oracle applications:
  - Inventory (INV)
  - Bill of Material (BOM)
  - Manufacturing (MFG)
  - Order Entry (OE)
  - Purchasing (PO)
  - Work in Process (WIP)
  - etc.



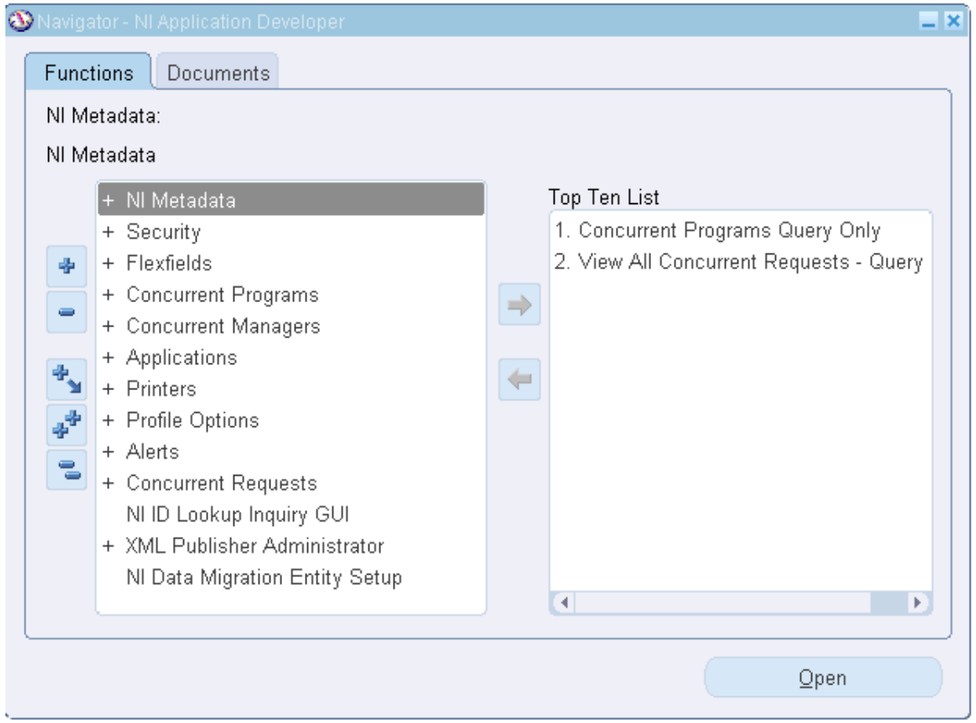

- Custom applications:
  - NIINV, NIBOM, NIMFG, NIOE, NIPO, NIWIP, NIMRP, NIMSC etc.

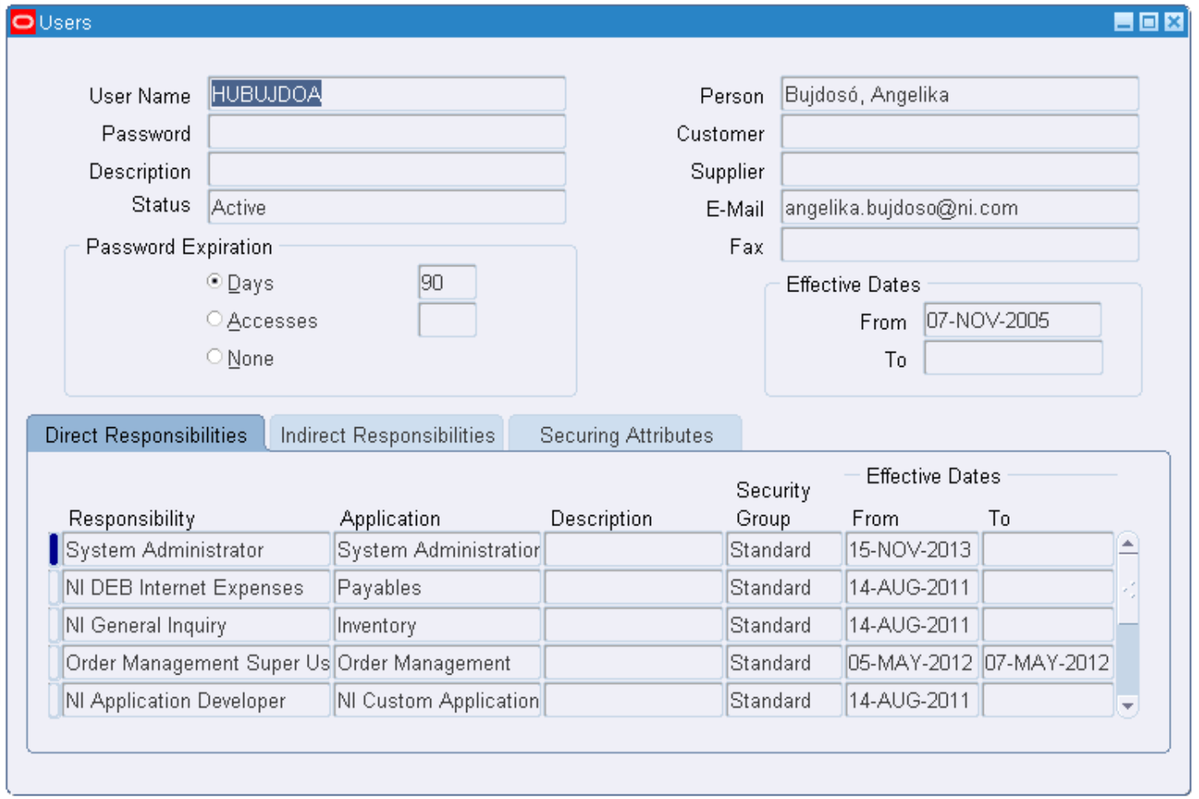
# Bevezetés az Oracle EBS világába

- ERP systems
- Oracle Applications
  - Logging in, menu, responsibility
  - GUI (LOV, flexfields, querying data etc.)
  - Modules (applications)
  - Concurrent managers
  - AOL objects
    Value set, Oracle Alert, Quick code, Flexfield, Concurrent program, Request set
  - How to add a new form to a menu (form, function, menu)
  - Accessible programs and forms for a user
    (user, responsibility, request group, menu)

# Menu, Responsibilities

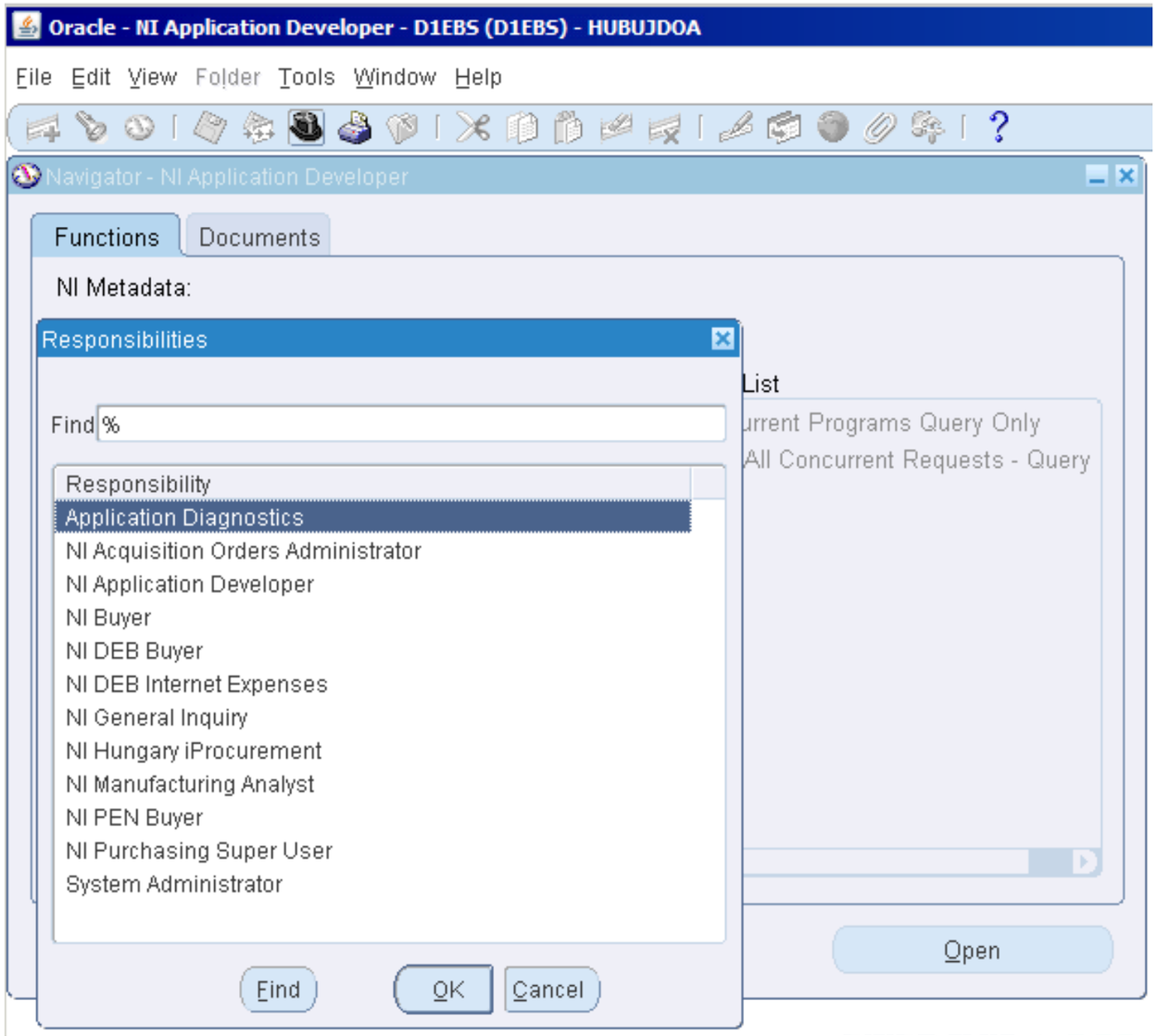
- Once you open a form a new menu window opens as well...



# Changing responsibility

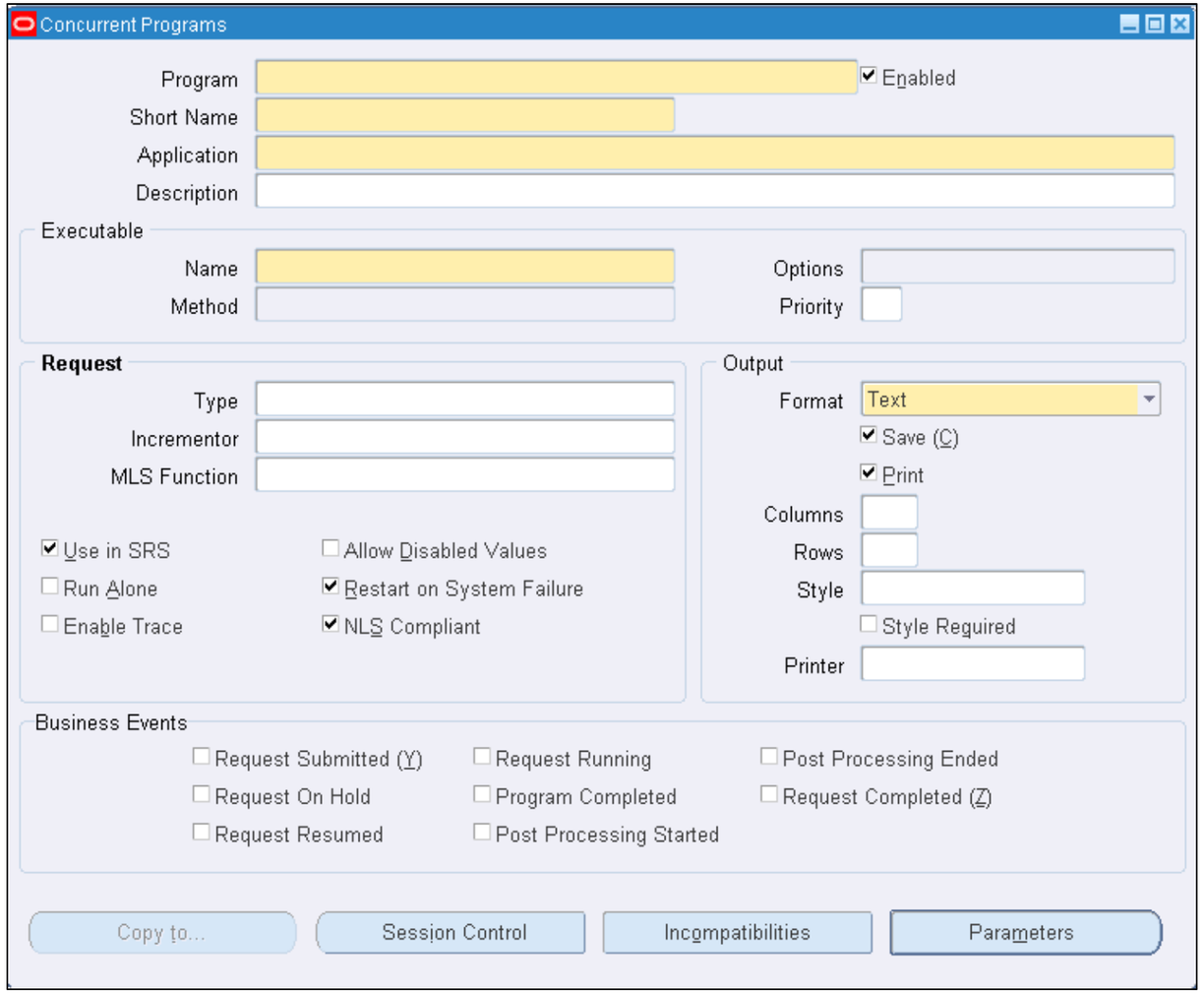
- Click on the top hat icon and choose another responsibility assigned to the current user



# Oracle Applications GUI

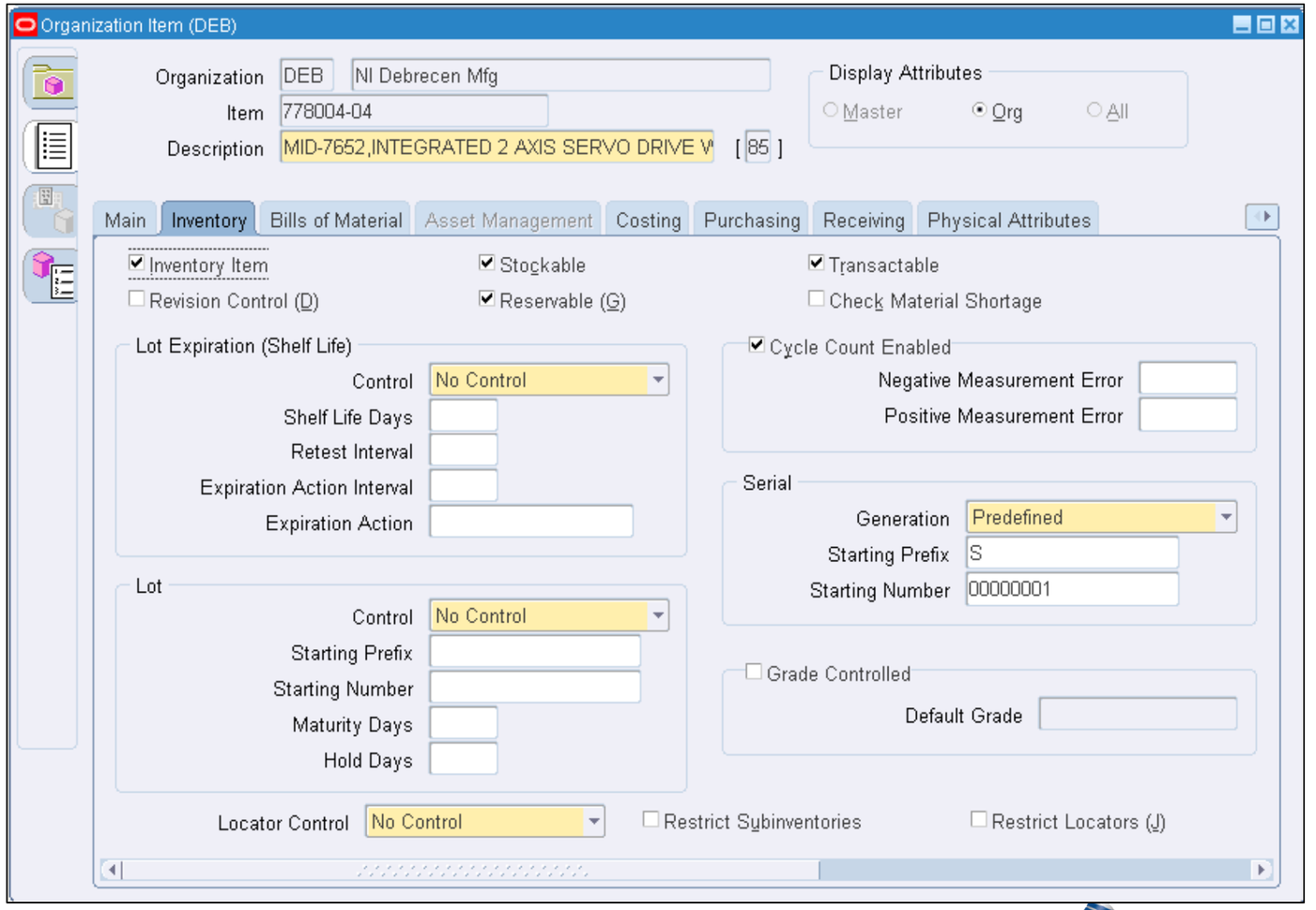
- Meaning of colors:

- Mandatory: yellow
- Optional: white
- Disabled: grey



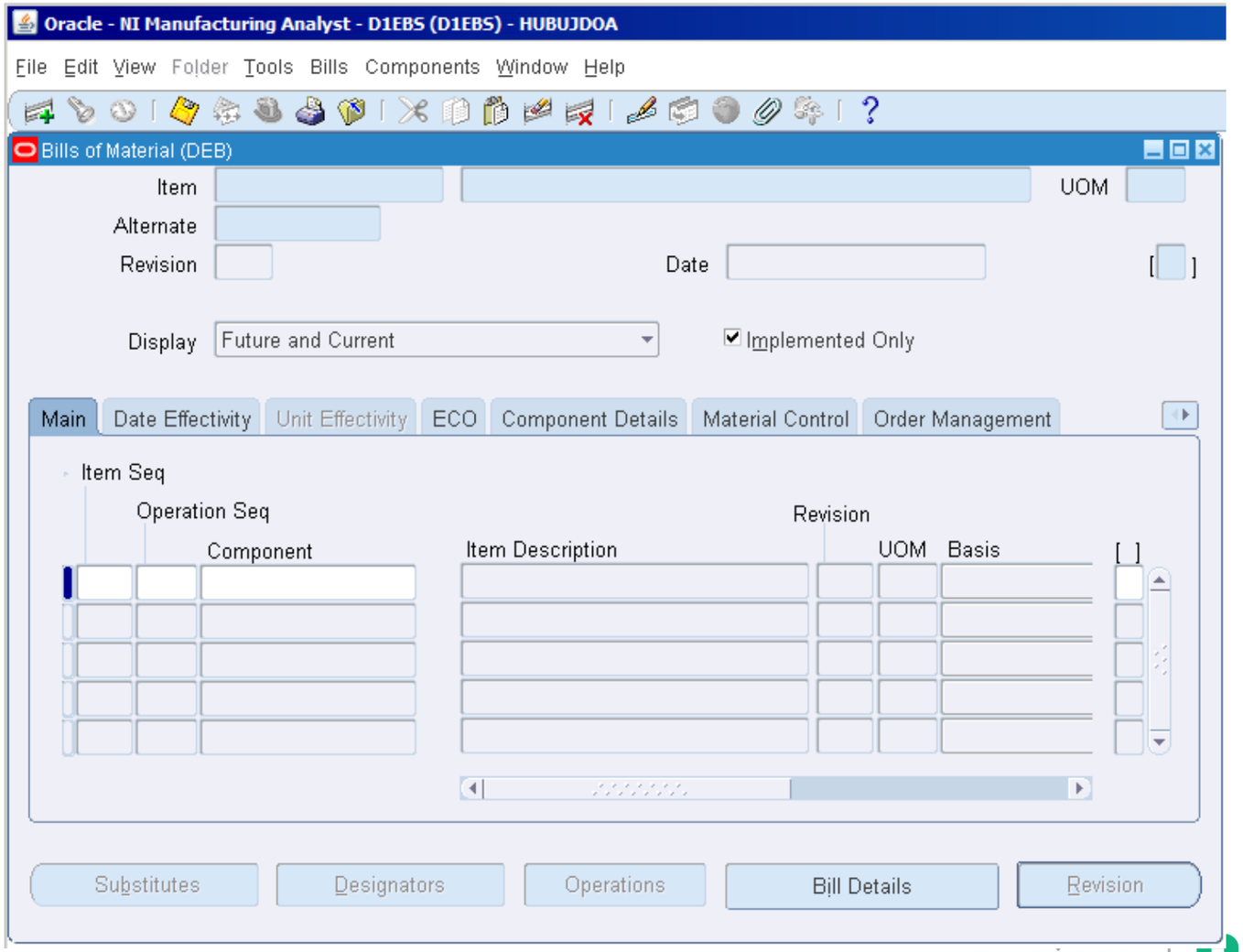
# Oracle Applications GUI

- User interface elements:
  - text boxes
  - buttons
  - checkboxes
  - radio buttons
  - tabs
  - scrollbars
  - list of values
  - combo boxes
  - etc.



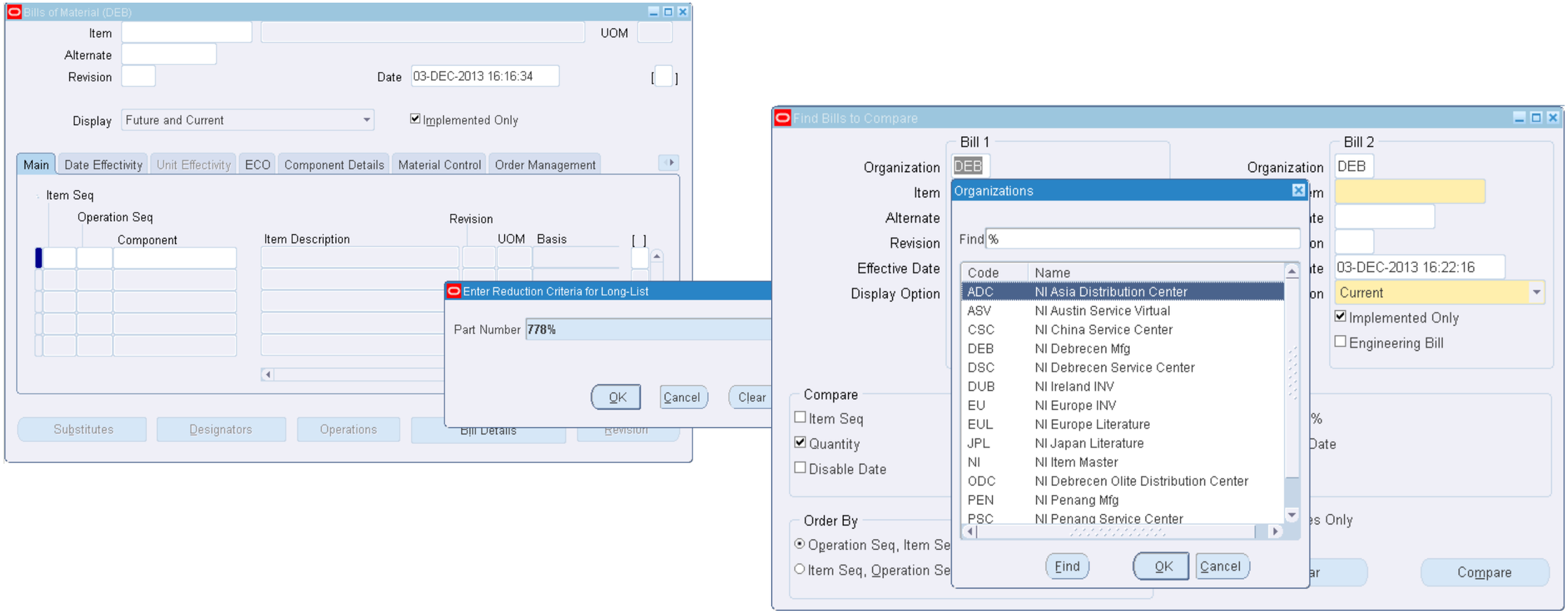
# Querying Data

- Entering „Enter Query” mode:
  - F11
- Executing the query:
  - CTRL + F11
- Leaving „Enter Query” mode:
  - F4

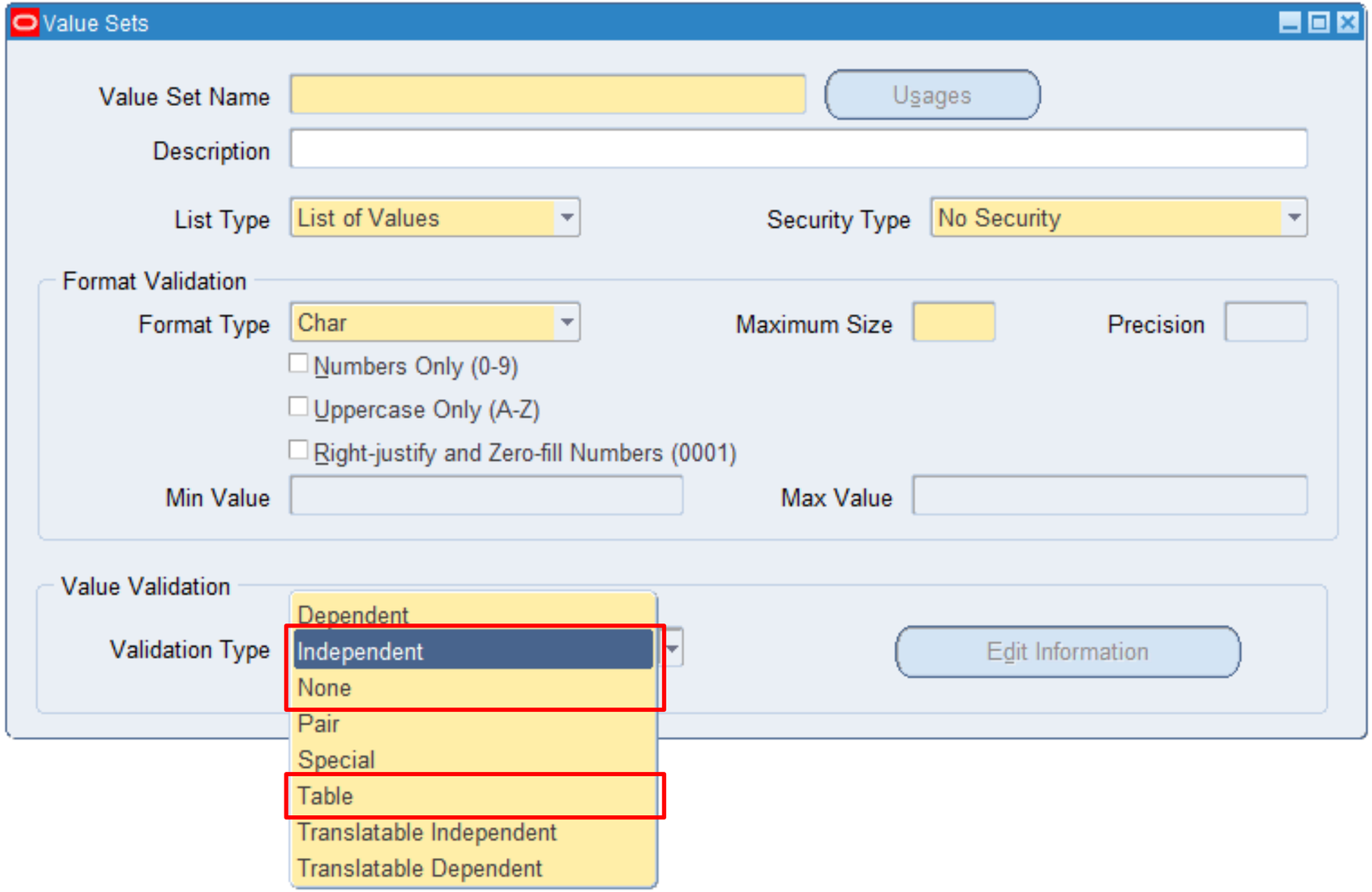


# List Of Values (LOV)

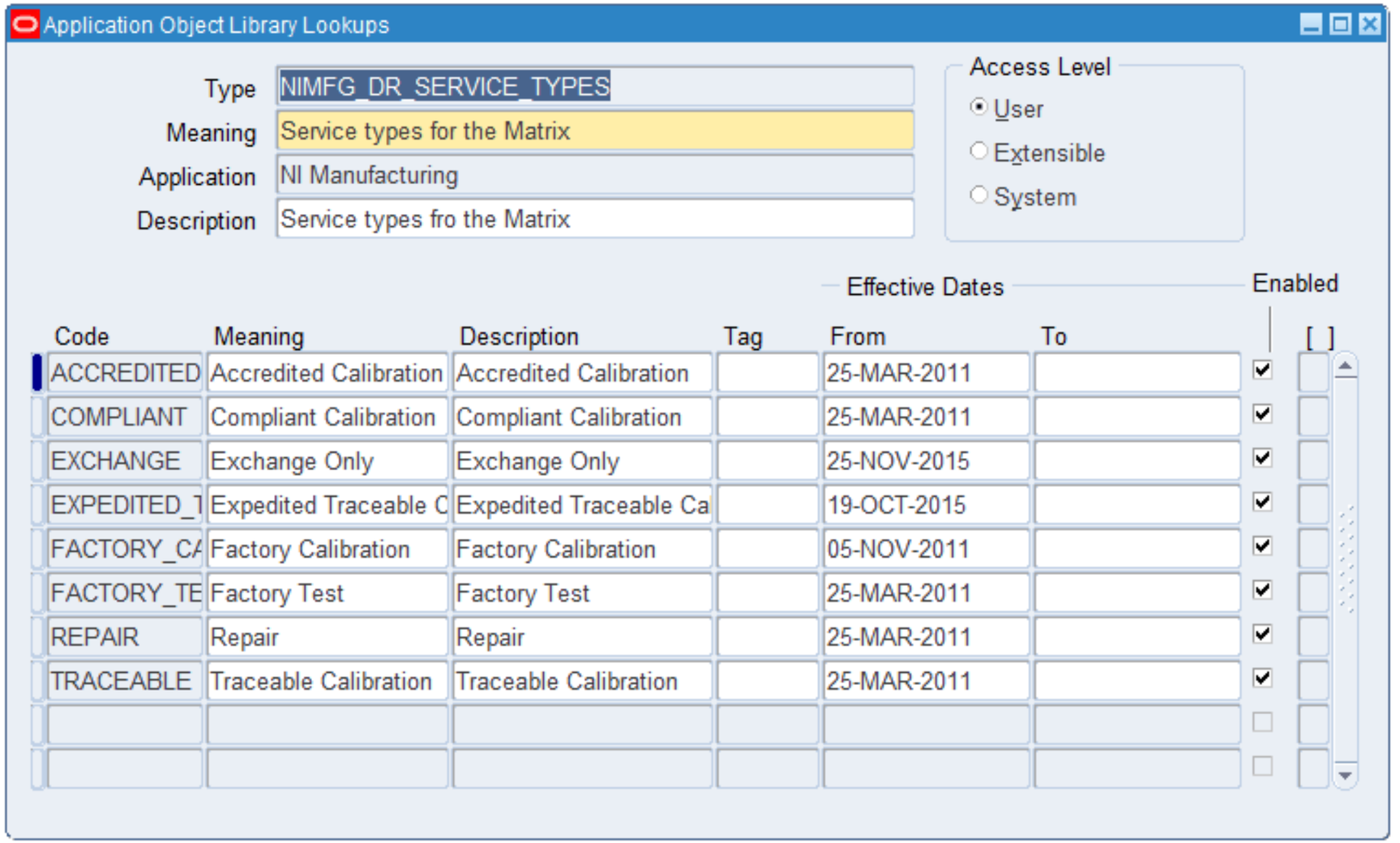
- Opening a list of values (LOV): clicking on the „…” button next to the field



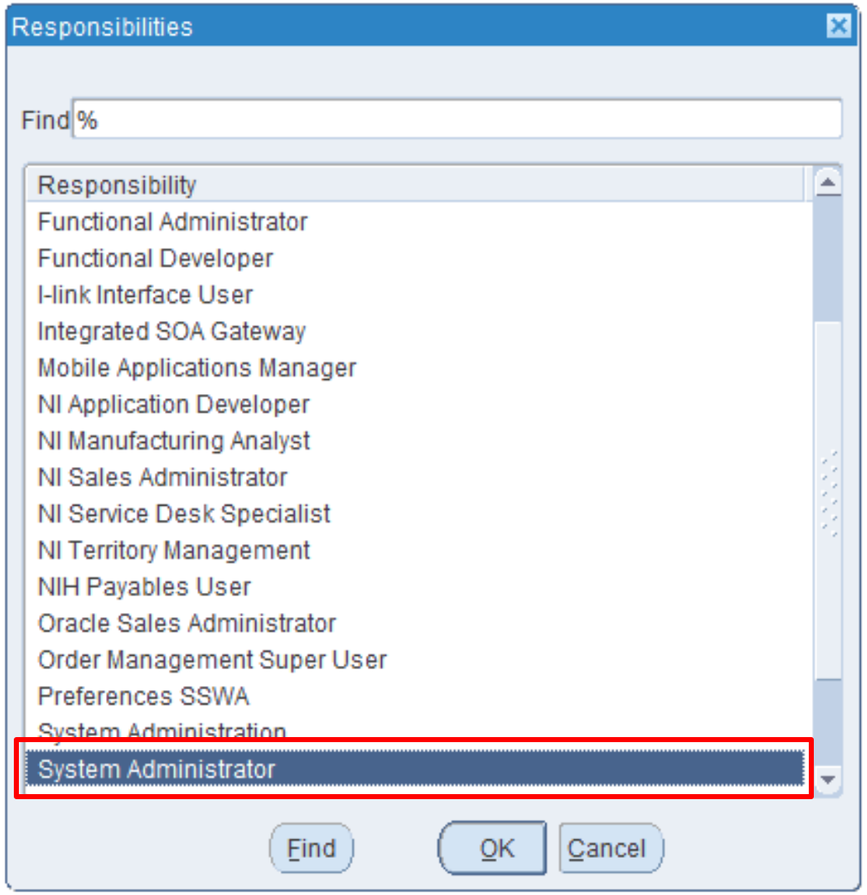
# Value Sets

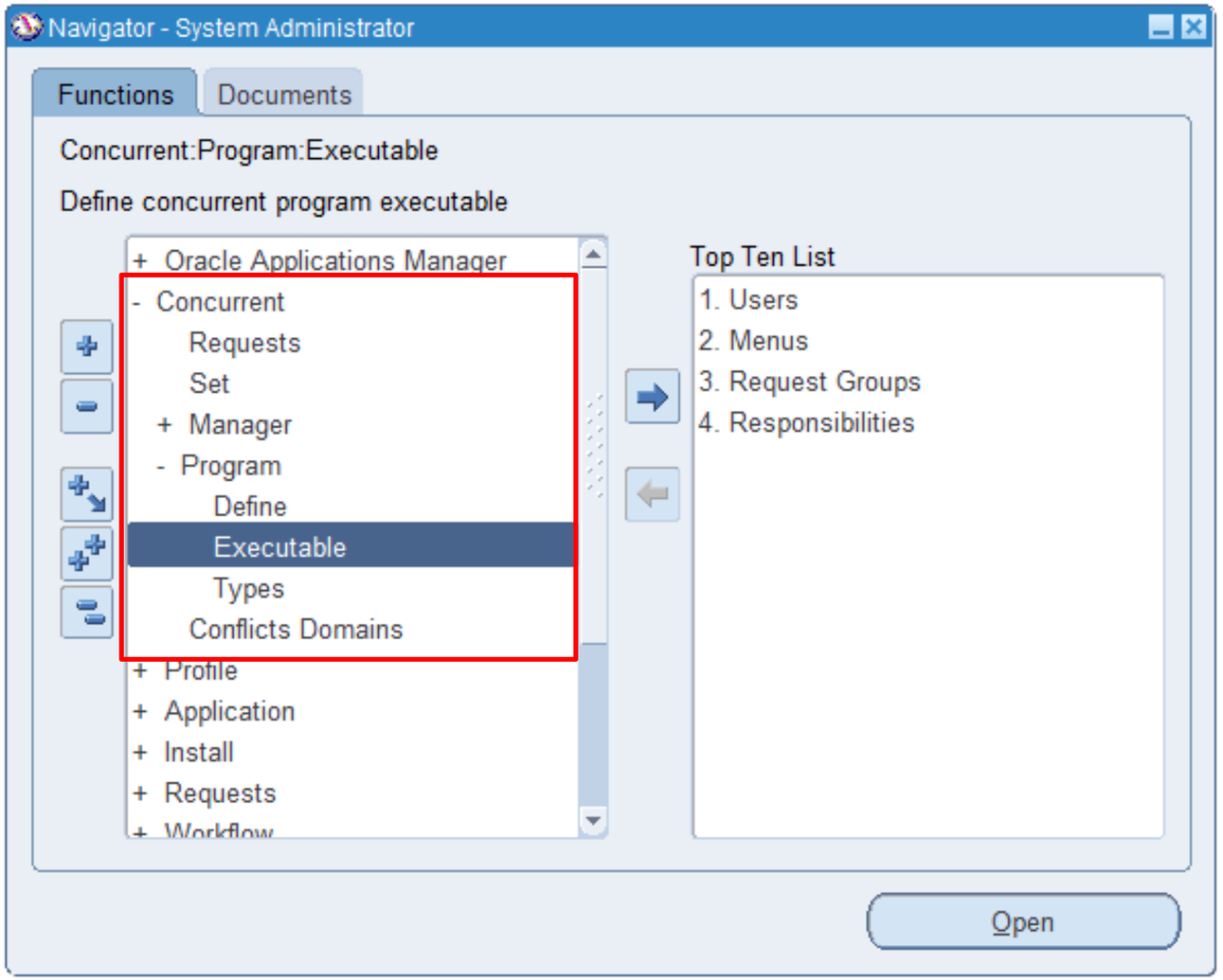


# Lookups



# Executables



# Concurrent Programs - Executable