ILP PROGRAM - ORACLE APPLICATIONS

Tata Consultancy Services

Oracle Reports Study Guide – Day2

Author:	MD. Nazmul Hoda (TCS)
Creation Date:	Jan 7, 2015
Last Updated:	Jan 7, 2015
Document Ref:	ILP/ORACLEAPPS/REPORT/Day2
Version:	DRAFT 1A

Approvals:

<Approver 1>	Santanu Sarkar (TCS)
<Approver 2>	Shubho Chakraborty(TCS)

Document Control

Change Record

Date	Author	Version	Change Reference
07-May-15	MD. Nazmul Hoda	Draft 1a	No Previous Document

Reviewers

Name	Position

Distribution

Copy No.	Name	Location
1	Library Master	Project Library
2		Project Manager
3		
4		

Note To Holders:

If you receive an electronic copy of this document and print it out, please write your name on the equivalent of the cover page, for document control purposes.

If you receive a hard copy of this document, please write your name on the front cover, for document control purposes.

Contents

Table of Contents

How to use this manual

This guide is continuation of “Oracle Reports Study Guide – Day1”. Continue reading it in the same way as Day1 guide.

1. Tools of Oracle Reports

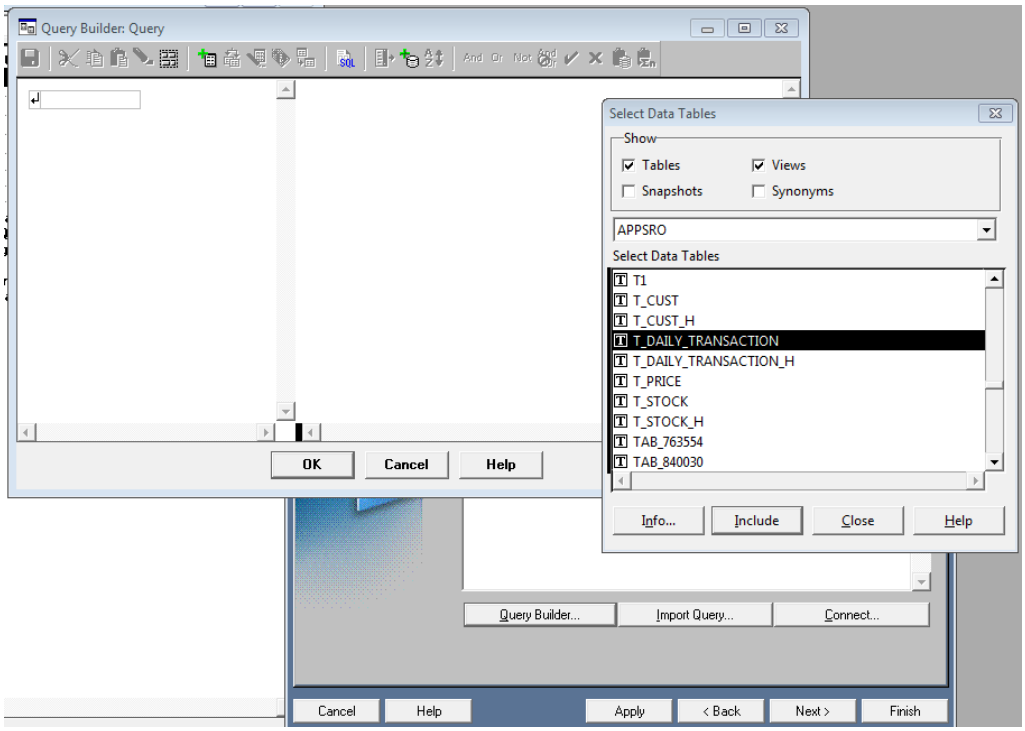
In this section you will learn how to use different tools available inside Oracle Reports to facilitate The development of reports

Now that we have learnt, how to build different kind of reports, let us look into the different tools available inside Oracle Reports.

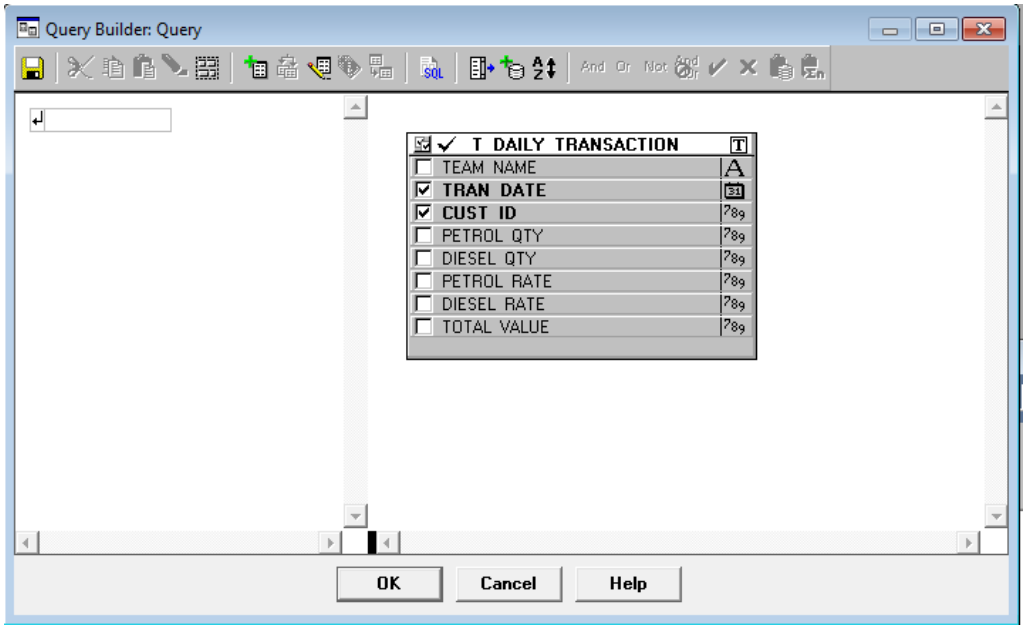
1.1 The Query Builder

The query builder helps in crating the sql query using mouse clicks. You can select the objects, the columns of object, the join and where conditions by mouse clicks. It also pulls out the column details of the object to help in selecting the columns. It crates the sql query for you.

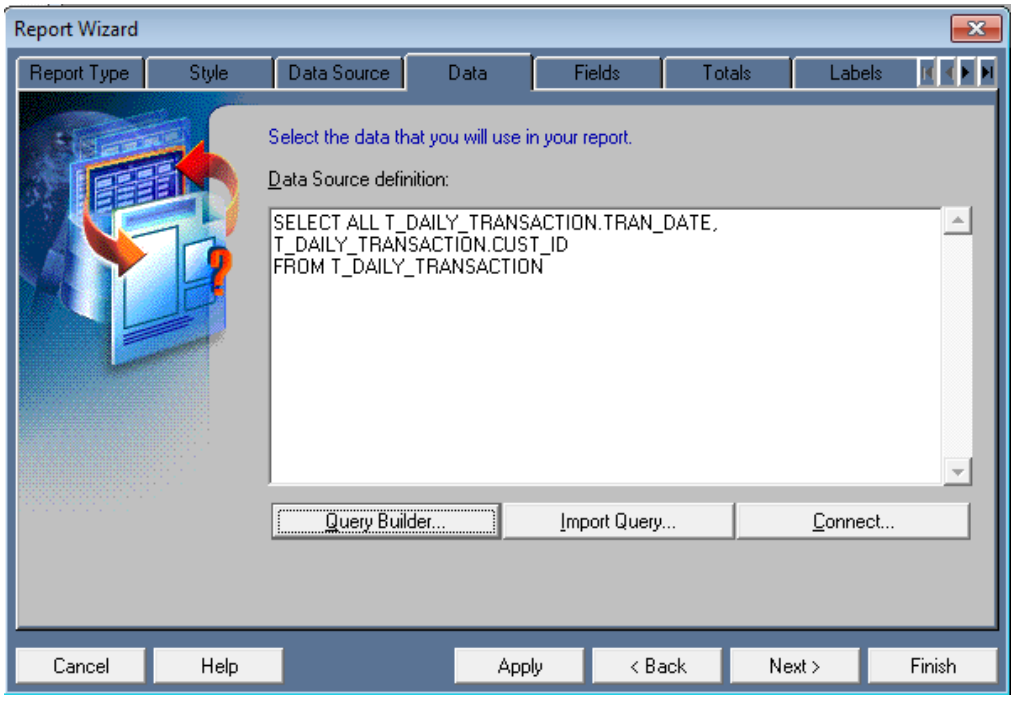
To start the query builder, click on the 'Query Builder' button on the report wizard, when you select the data source as 'SQL Query'. In the next window, you can select the Table/View/Snapshot /Synonym names by typing or selecting the name.



Click 'Include' (you can select as many objects you want), you will get the option to select the columns, and the conditions

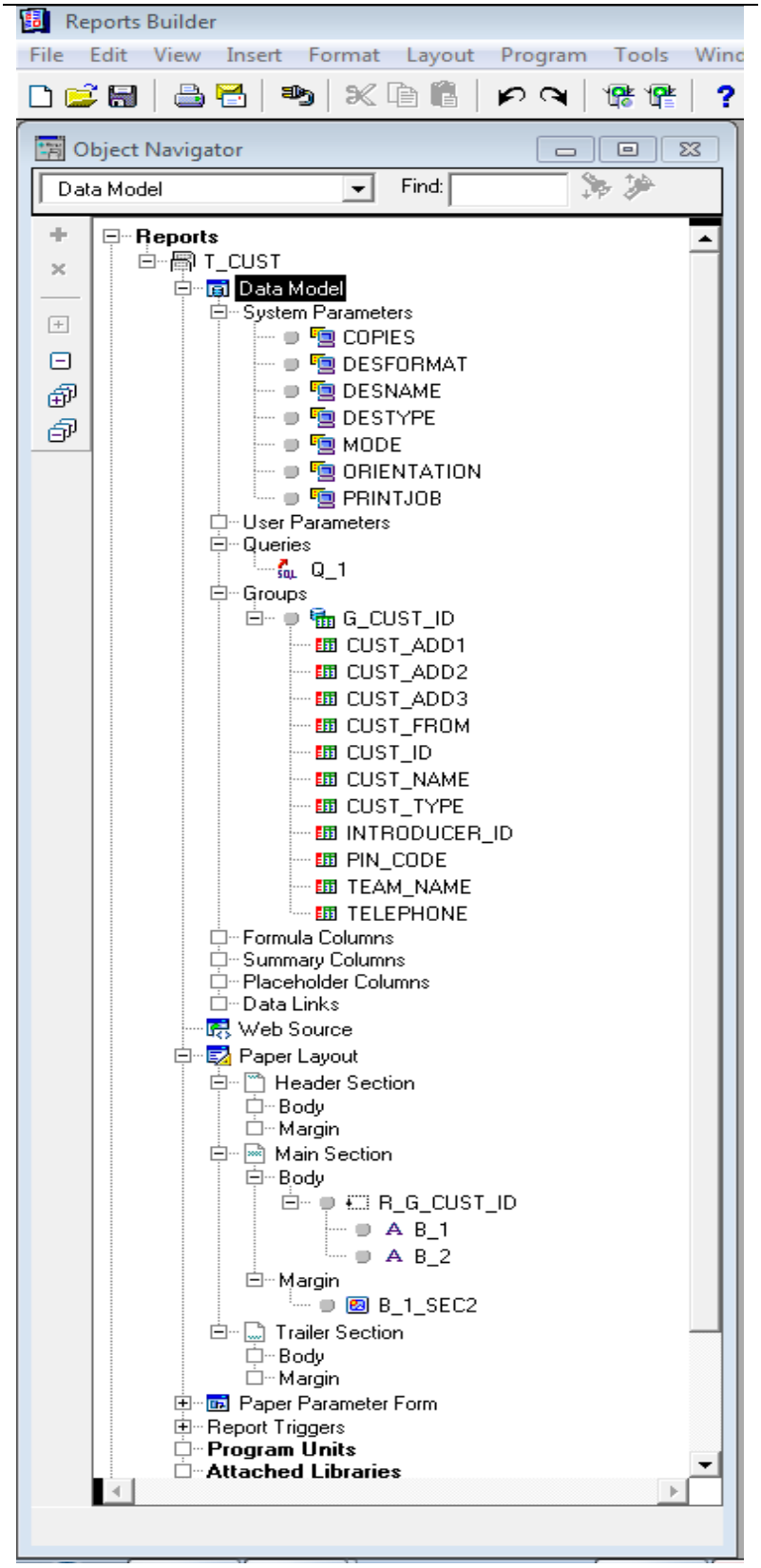


Clicking OK will create the sql query for you. Now you may proceed with the normal report building process.



Note: You can also pull the query from an existing file in your PC.

1.2 The Object Navigator



Object Navigator is the main console through which you can see the details of each component of the report. Here is the expanded view of the object navigator for our first sample report crated on table T_CUST.

We will carry put an explanation of each relevant section of the Object Navigator. Some sections will be covered in detail, and some in brief, depending on its applicability in our context.

1. **T_CUST** : Title of the report, you can give any name you like
2. **Data Model** This section provides a pictorial view of your queries and the associated data groups.

    a. **System Parameters**
        System defined parameters like orientation, copies etc.
    b. **User parameters**
        You can define your runtime parameters, you can use them inside your query and report layout also.
    c. **Query**
        We know this already
    d. **Group**
        One group is crated for each query, which holds the objects of the group, here group named G_CUST_ID has been created.
    e. **Formula Columns**
        These are the fields which can hold a value based on your calculation, for example, you many like to display a text 'Old customer' for the customers were CUST_FROM is < 01-JAN-1980
    f. **Summary Columns**
        These are the calculated columns – like count of customer IDs.
    g. **Placeholder Columns**
        It is kind of a variable, which can store the result of a computation, and can be used or referred back at other places.
    h. **Data Links**
        A link is created to create relationship between two groups, we will understand the details in the later sections.

3. **Paper Layout -** When you click on this, it opens a Layout Editor, where you can organize your objects on how they will appear on the output. You can drag the objects to the position you want. This has following sections which is self-explanatory. When you click on them, it opens the 'Property Inspector' window which shows the details of the object's property (like Length, width etc.).

    a. **Header Section**
        i. **Body**
        ii. **Margin**
    b. **Main Section**
        i. **Body**
        ii. **Margin**
    c. **Trailer Section**

4. **Paper Parameter Form** This gives the layout of the parameter form (if used), which will pop-up at runtime and you can capture the parameters and pass it to the query or use inside the formula columns etc. You can define the look and feel of the fields appearing in the parameter window here.

    a. **Fields** Field in the parameter for
    b. **Graphical Boilerplate** Any fixed graphics you intend to put
    c. **Text Boilerplate** Any fixed text you intend to put
    d. **Image Boilerplate** Any image graphics you intend to put

5. **Report Triggers** If you want to do any preprocessing before the report starts (or at the events mentioned below), the following triggers are available to be used. The name is self-explanatory.

    a. **BEFORE PARAMETER FORM**
    b. **AFTER PARAMETER FORM**
    c. **BEFORE REPROT**
    d. **BETWEEN PAGES**
    e. **AFTER REPORT**

6. **Program Units**

    You can define package, procedure and function to here, which can be used within the report to perform a specific function. This is useful when a particular calculation or action is done at many places inside the report, then you can define it inside a program unit and call wherever needed.

7. **Attached Libraries**

    You can attach a library (collection of functions, codes) into the report, and can refer to the function/producers defined within the library.

8. **Templates**

    Templates define common characteristics and objects that you want to apply to multiple reports. For example, you can define a template that includes the company logo and sets fonts and colors for selected areas of a report. You can use a ready template or create your own template. For Example if your project is creating 100 reports for company ABC Corp, it is better to use a common template for all the reports, so that look and feel of all the reports are same.

9. **PL/SQL Libraries**

    If you have many reports that use these same ref cursor types and SELECT statements, you can move the program units that you created into a PL/SQL library stored in a file or the database, so that other reports can easily share the code.

10. **Debug Action**
    The PL/SQL Interpreter can be invoked from report code (triggers, user-named program units, libraries, and so on) by creating debug actions in the code. These are instructions that track the execution of PL/SQL program units so they can be monitored. This will be discussed in detail later, for now just remember that this tool helps you in debugging the code by stopping it at particular place during runtime and examining the values at that moment

11. **Stack**

    Not important for current discussion.

12. **Built-In Packages**

    Reports Builder is shipped with a built-in package (SRW), a collection of PL/SQL constructs that include many functions, procedures, and exceptions you can reference in any of your libraries or reports. The PL/SQL provided by the SRW package enables you to perform such actions as change the formatting of fields, run reports from within other reports, create customized messages to display in the event of report error, and execute SQL statements.

13. Database Objects

Not important for current discussion.

Video5 Script: Vid6-Rep-Tools.avi (Screenplay of creating a Matrix report)

This video will show the screen capture of different tools available within Oracle Developer reports, and explains each of them with their usage and examples

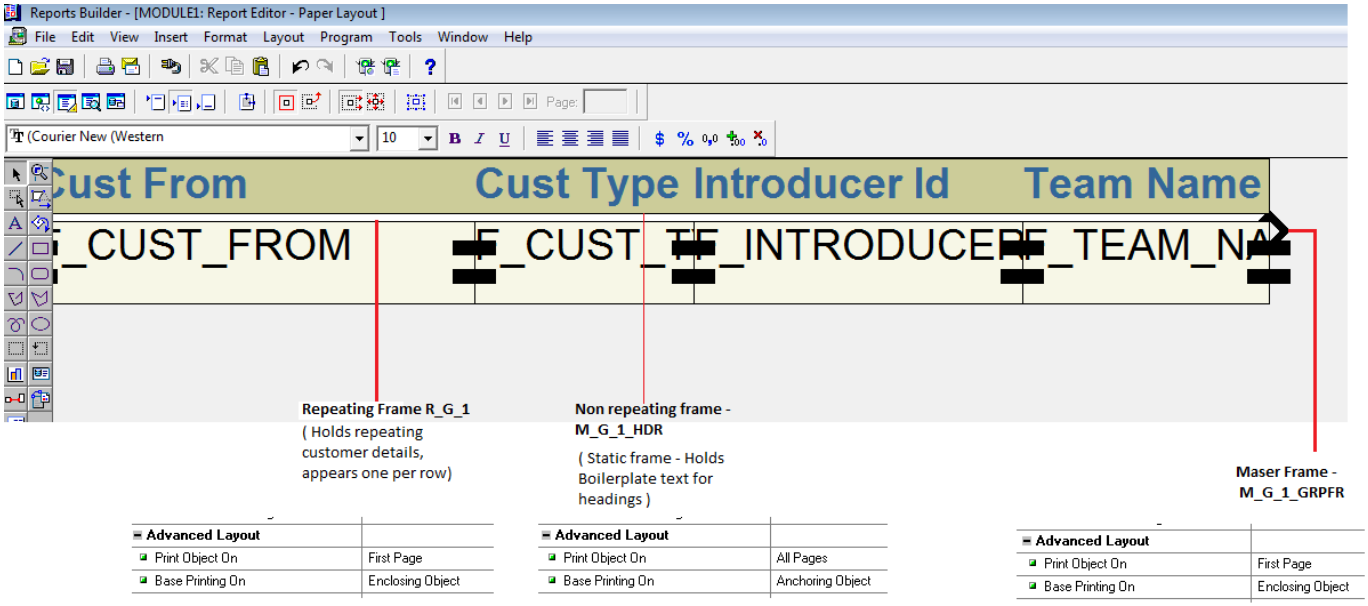
1.3 The Layout Editor

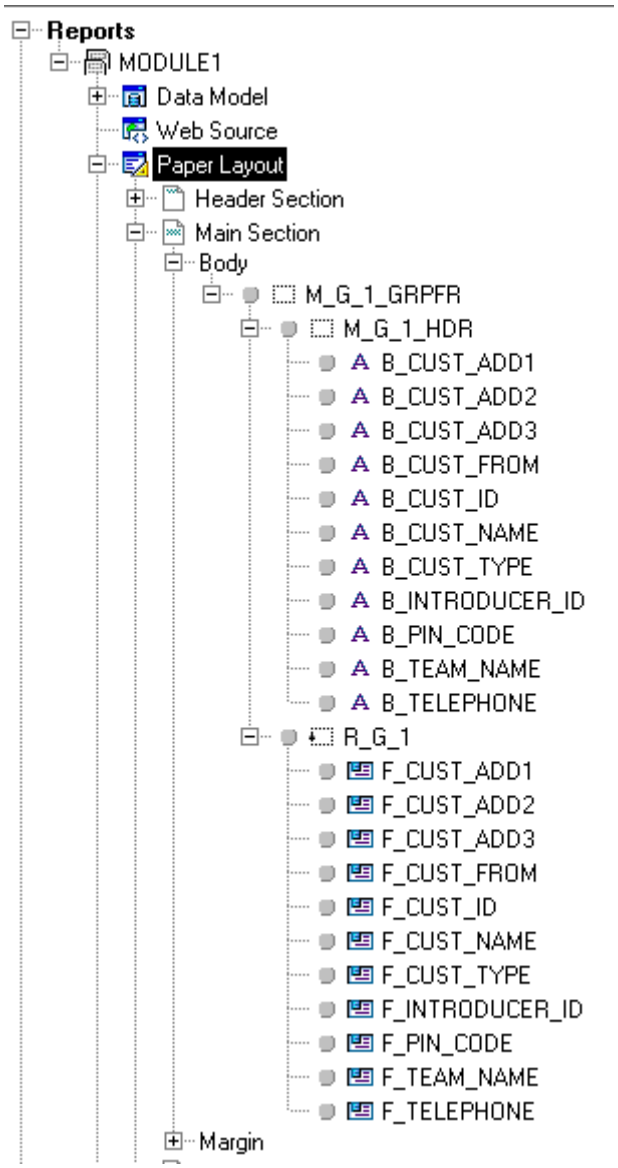
In this section you will learn about the structure of the frames and groups

Let us understand different features of the layout editor. We will take a new table – EMPLOYEES for this example.

Layout editor is used to edit the look and feel of the report output, and arrange the fields of on the report at specific places. To go to the layout editor, click the button before the Layout Editor in the object navigator screen. This will open the layout editor screen.

This is the layout for the very first report we have developed. Revisit the section and try to correlate the3 different frames appearing in the layout editor. Also look at the expanded mode of the layout editor for this report. Look at the groups and the frames and correlate them with the Property of the frames shown in the below diagram.





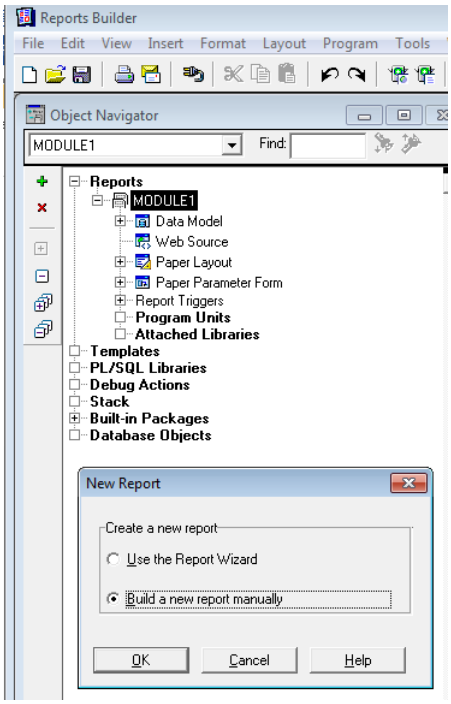
You can double click on the button left of the object to open the property inspector window, showing all its detail.

2. Program Units - Manual Data Model

So far, we have been using the Report Wizard to create a report. In this section, we will see how to create a report manually

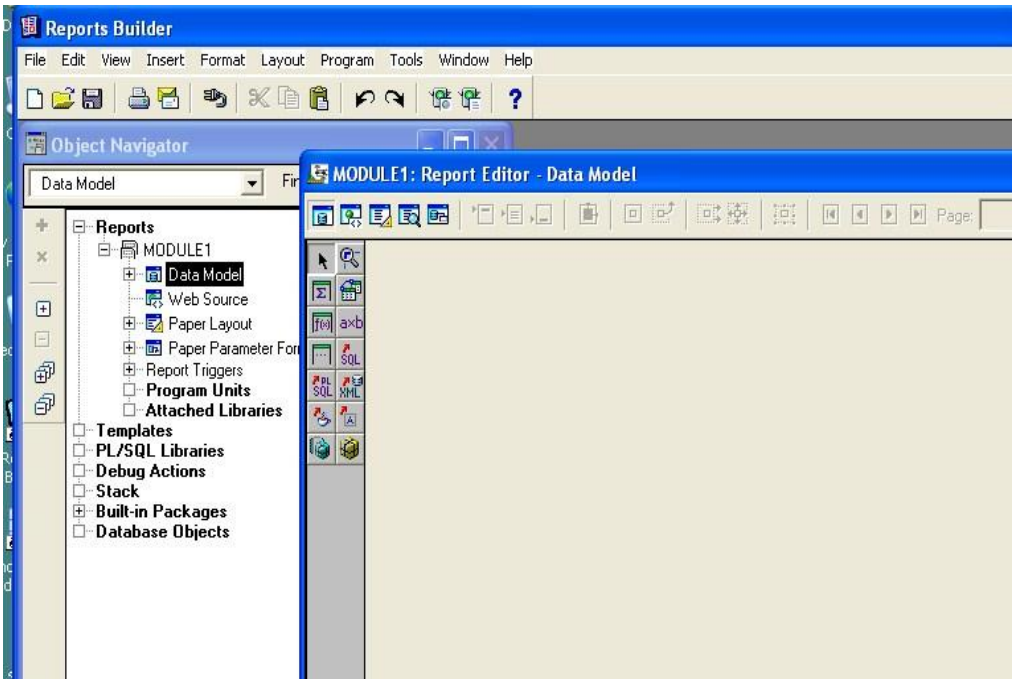
Open the Report Builder & Login as appsro user. We will be using a table – EMPLOYEE, if the table does not exist, use the script on in Appendix – A (Pg. 79) to create the table, and insert data into it.

Place the mouse on Data Model Option and double click, then



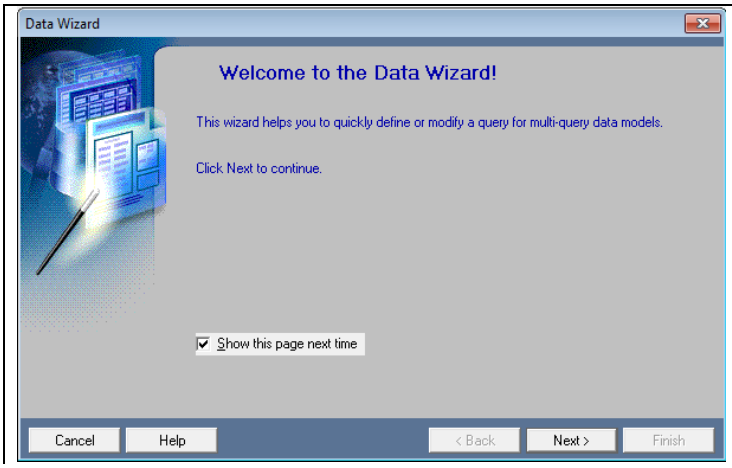
Open the report builder and click File > New from the toolbar, you will be prompted to select from 'Report Wizard' or 'Manual'.

Select 'Manual'

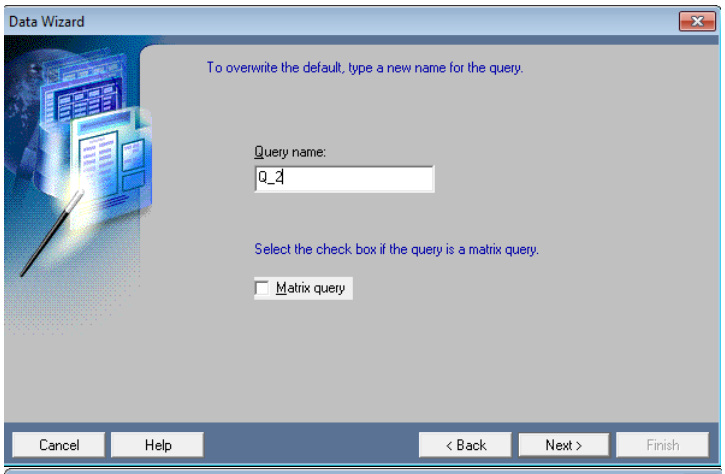


The System bring it into blank data model screen

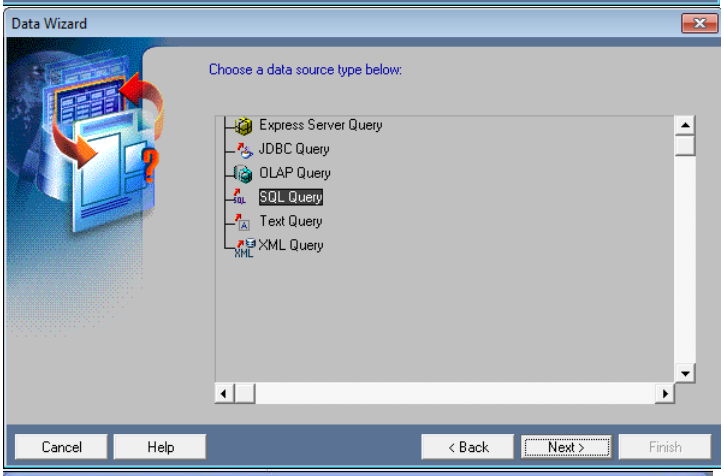
This is the full Blank Data Model Screen. As a programmatic approach, variety of tools available on left side. As this stage to double click on SQL icon and place on blank model session, then system automatically open the Query Builder tool



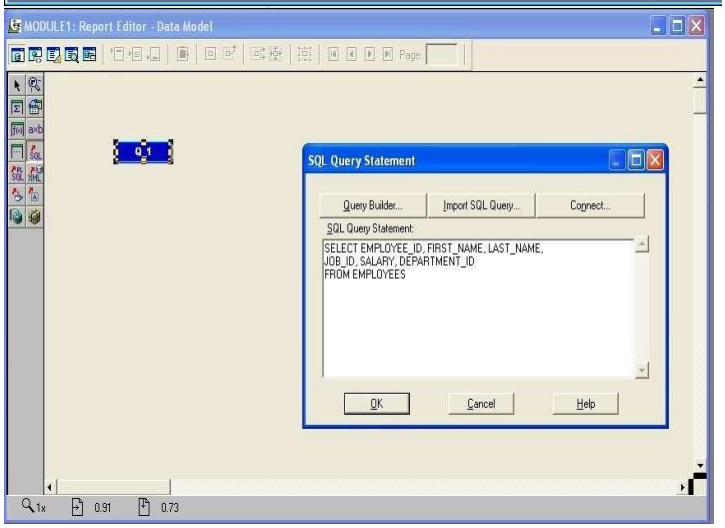
Right click and select Data Wizard, Select Next



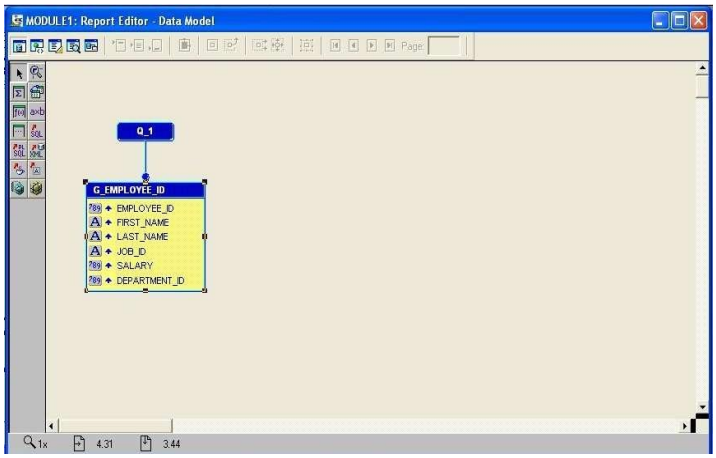
Supply the name of query, Select Next



Select 'SQL Query' as data source, click Next



Supply the SQL Query, this will create a small box for the query inside the editor.

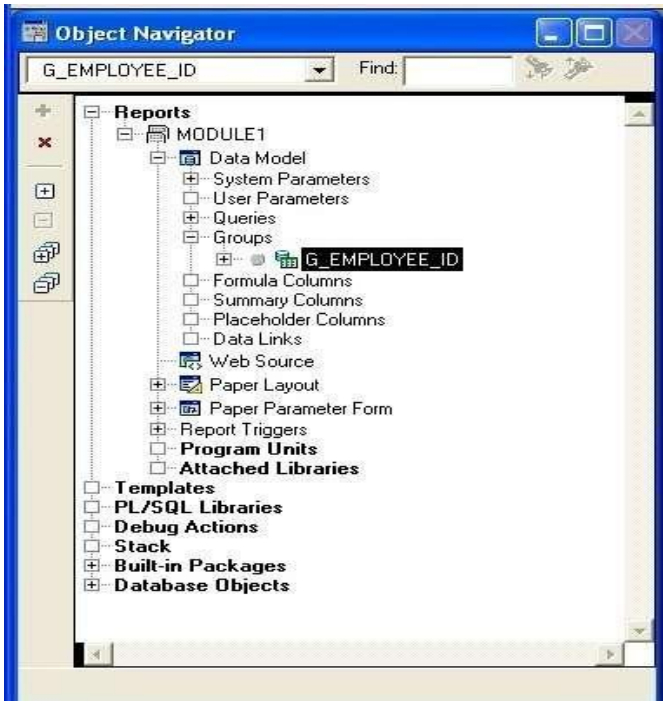


In the next screen, do not select any columns for 'Group Field', click Next.

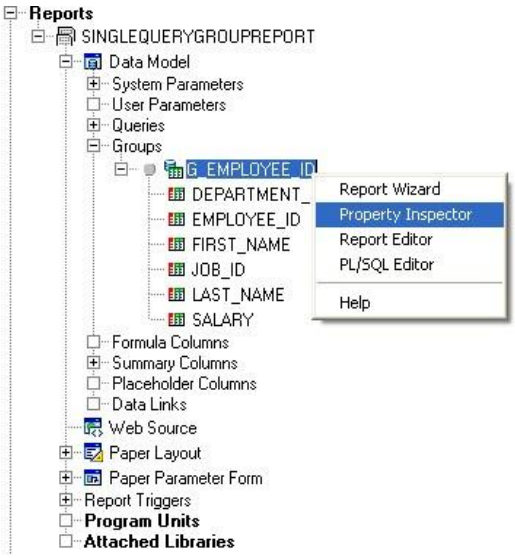
Also skip the 'calculate total' screen and hit 'Finish'.

This will create a group named G_EMPLOYEE_ID, which will show the columns contained in the group.

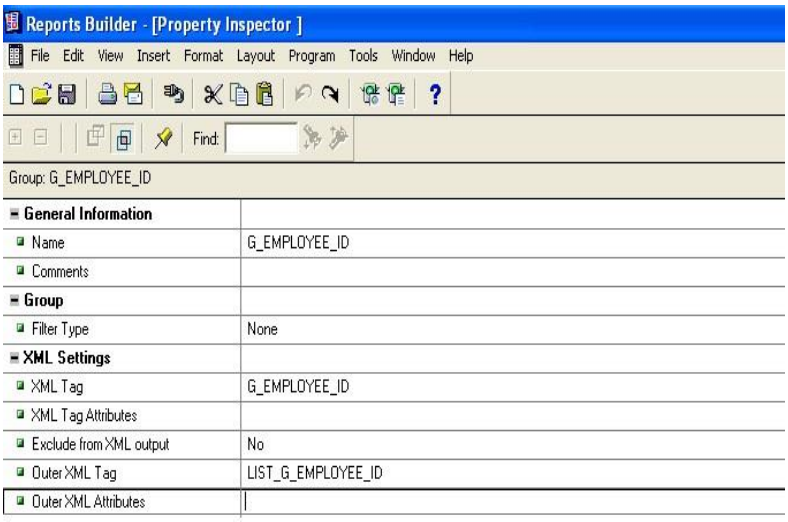
On Object navigator the Group is placed on Groups session which is the sub-section of Data Model. Now *Right* click on the Group name (G_employee_ID). *Click* on + button will expand the Group and then *Right Click* it, will show the selection option list



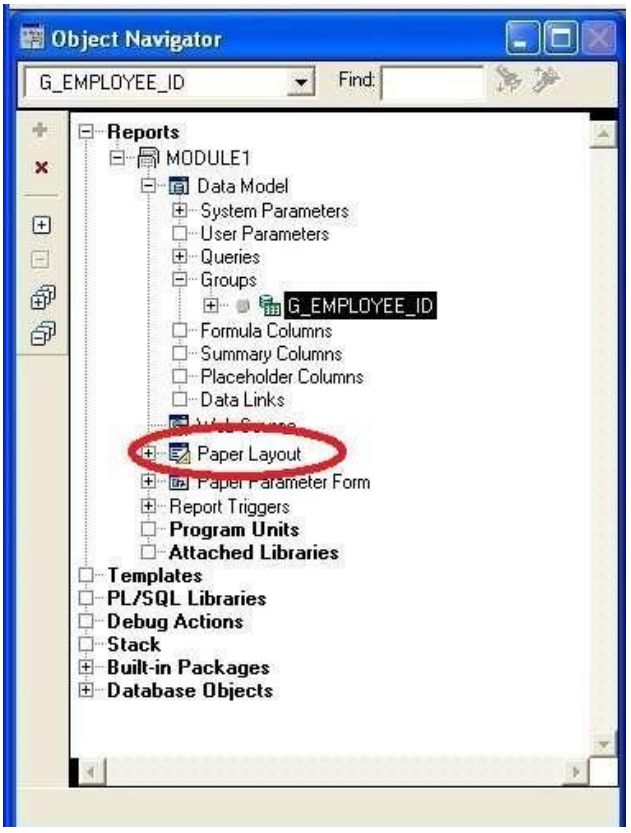
Then select the *Property Inspector* will show the property panel.



The Property Panel Details as Below:

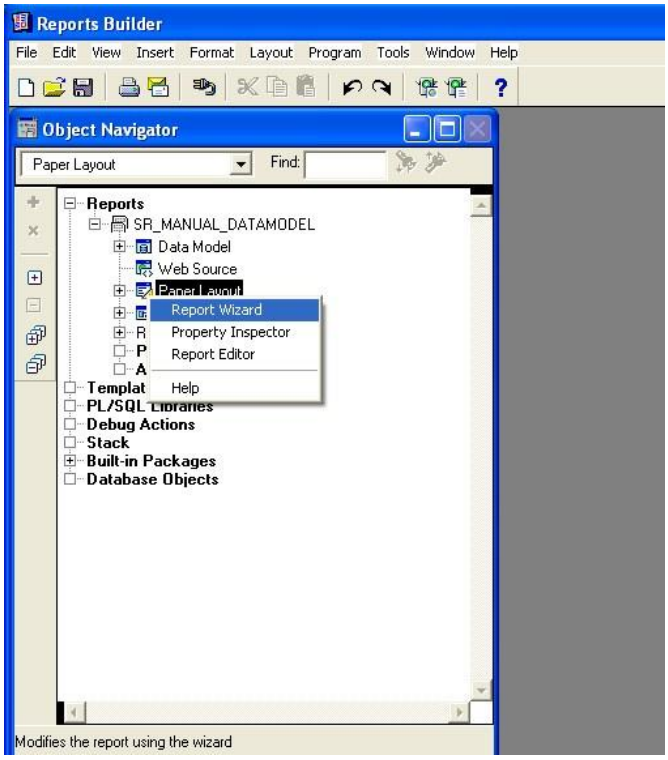


The Name option value is the Group Field. If you need to change then alter the value. Whenever change *Name* Value then copy the same value to *"XML Tag"*

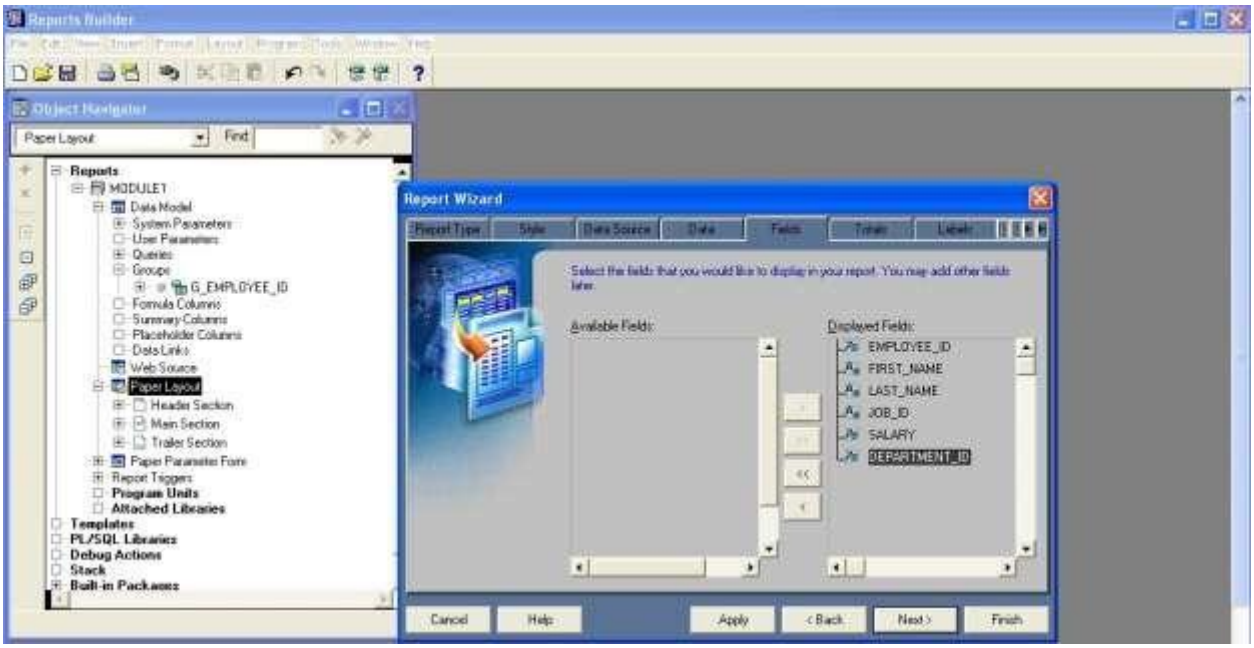


Now the Next task is to create a Layout using Manual option. For the purpose, place the mouse on Paper Layout then *Right Click* it, will show the selection option list

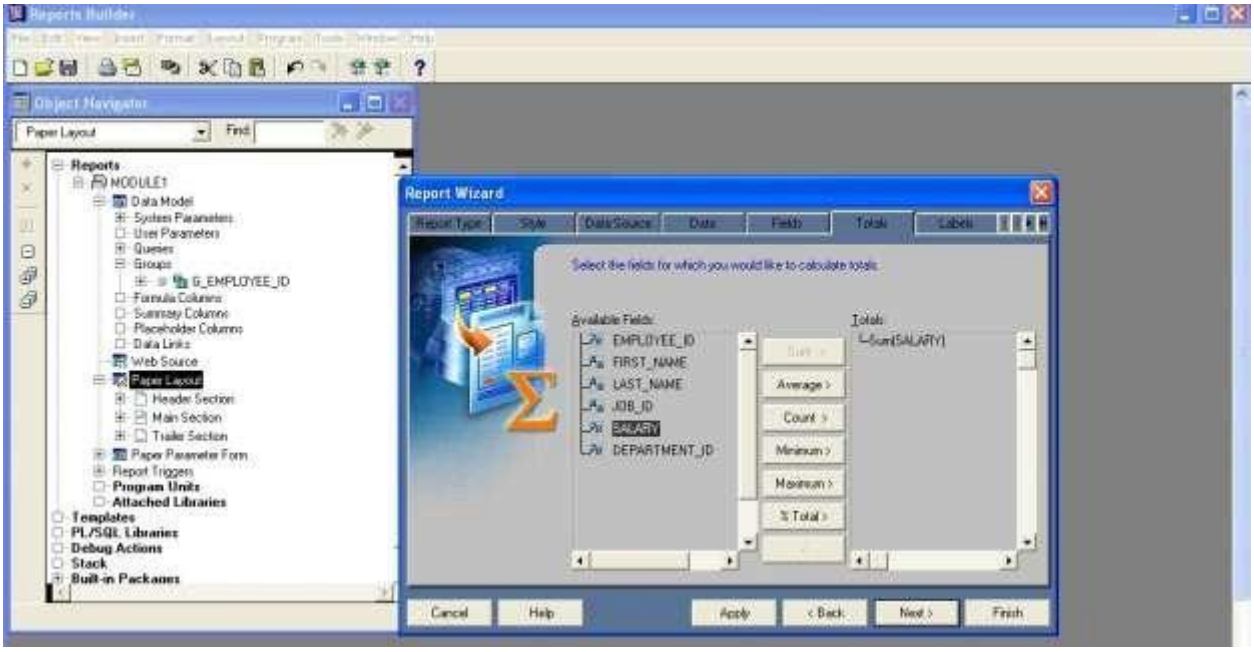
Then select the *Report wizard* Option and follow the wizard's Guide Flow path



Select the fields as you need on report screen. As per earlier discussion and even all demo, the Available fields and Displayed Fields are explained. If you need recalling the explanation

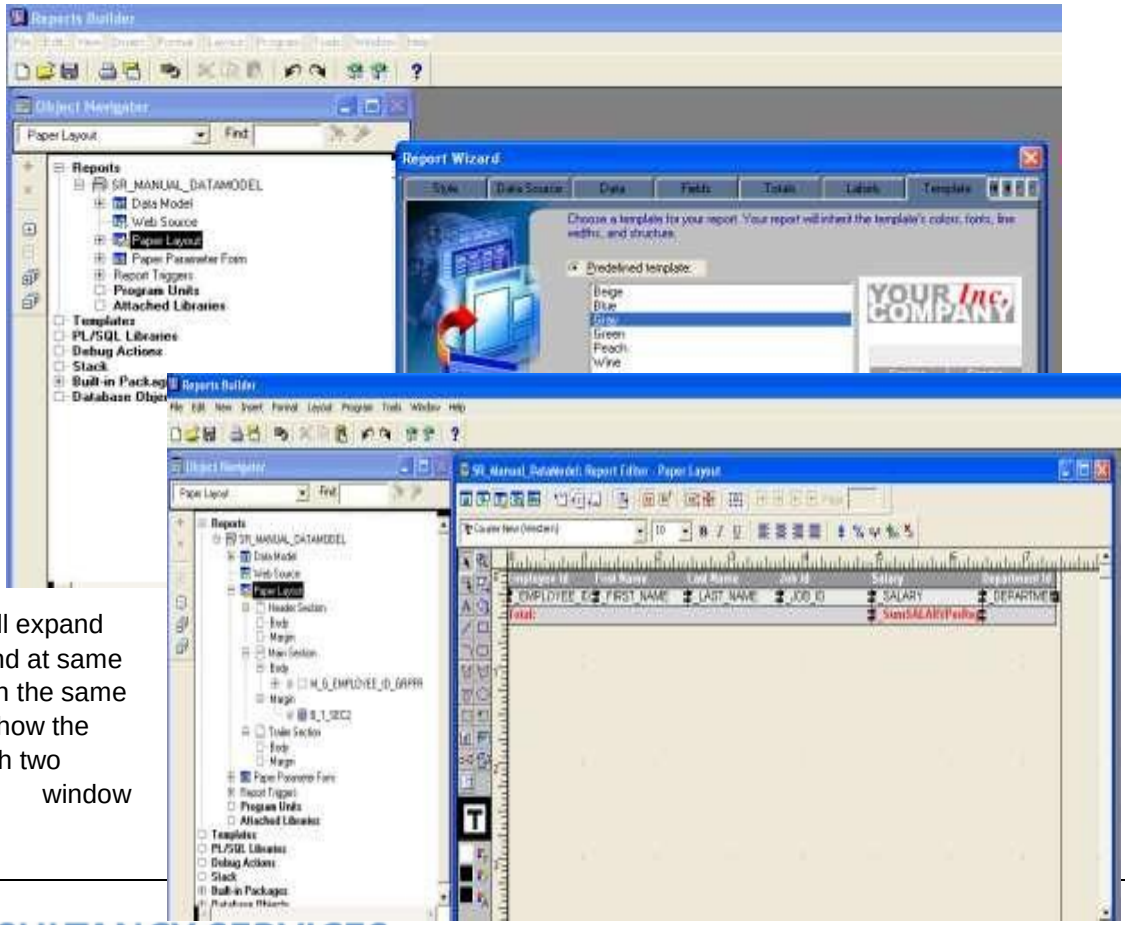


Then for calculate the sum based on Salary to use the Sum value Calculation Screen,



Finally select the corresponding Default Layout Templates

Once the List layout select is complete and comeback on the Object Navigator.



Then *Click* on + button will expand the Paper Layout and at same time Double *Click* on the same Paper Layout, will show the screen as below with two session as single window

Navigator

The final output of the process will look like this:

Employee Id	First Name	Last Name	Job Id	Salary	Department Id
100	Steven	King	AD_PRES	24000	90
101	Neena	Kochhar	AD_VP	17000	90
102	Lex	De Haan	AD_VP	17000	90
103	Alexander	Hunold	IT_PROG	9000	60
104	Bruce	Ernst	IT_PROG	6000	60
105	David	Austin	IT_PROG	4800	60
106	Valli	Pataballa	IT_PROG	4800	60
107	Diana	Lorentz	IT_PROG	4200	60
108	Nancy	Greenberg	FI_MGR	12000	100
109	Daniel	Faviet	FI_ACCOUNT	9000	100
110	John	Chen	FI_ACCOUNT	8200	100
111	Ismael	Sciarra	FI_ACCOUNT	7700	100
112	Jose Manuel	Urman	FI_ACCOUNT	7800	100
113	Luis	Popp	FI_ACCOUNT	6900	100
114	Den	Raphaely	PU_MAN	11000	30
115	Alexander	Khoo	PU_CLERK	3100	30
116	Shelli	Baida	PU_CLERK	2900	30
117	Sigal	Tobias	PU_CLERK	2800	30

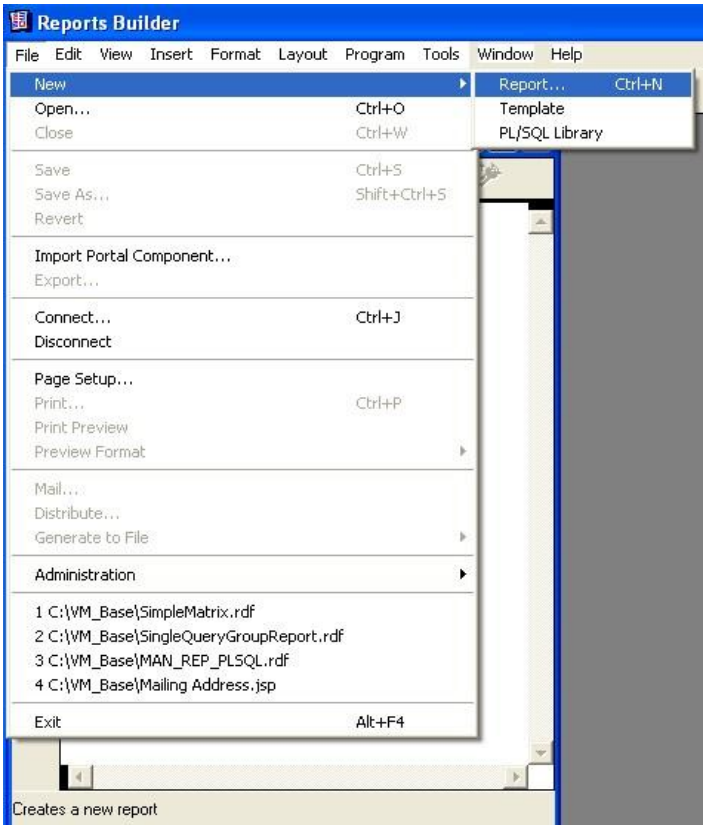
3. Program Units - PL/SQL Libraries

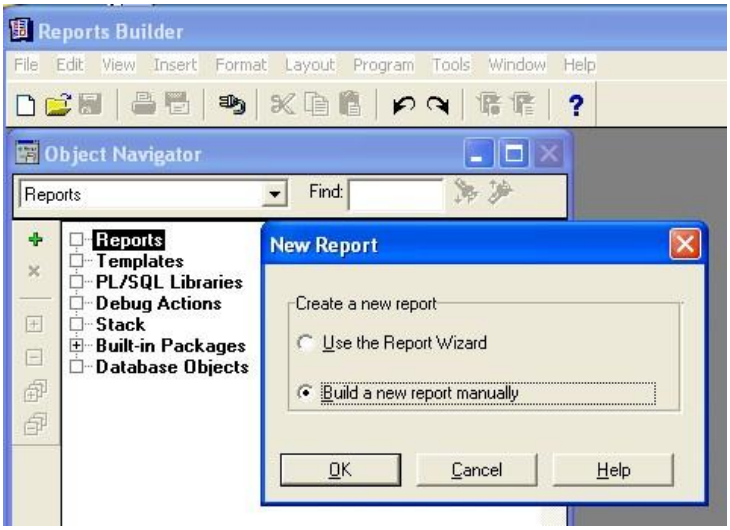
In this section you will learn how to create and use a PL/SQL library.

When we need to use a particular piece of code at multiple places, defining it in form of a PL/SQL library gives use several advantages:

(i) Duplication of code is avoided, you need to write less code
(ii) Code is centralized, so any change in the function will be reflected to all the places where it has been used.
(iii) It can be saved as a file and be used in multiple reports.

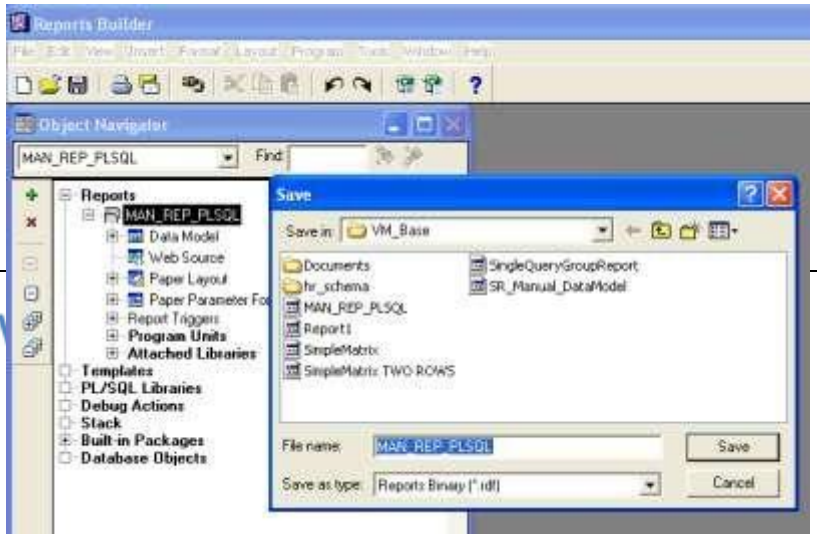
In the below example, we will create a function inside a PL/SQL library, which computes the Bonus, and then will call it inside the report to calculate the Bonus.





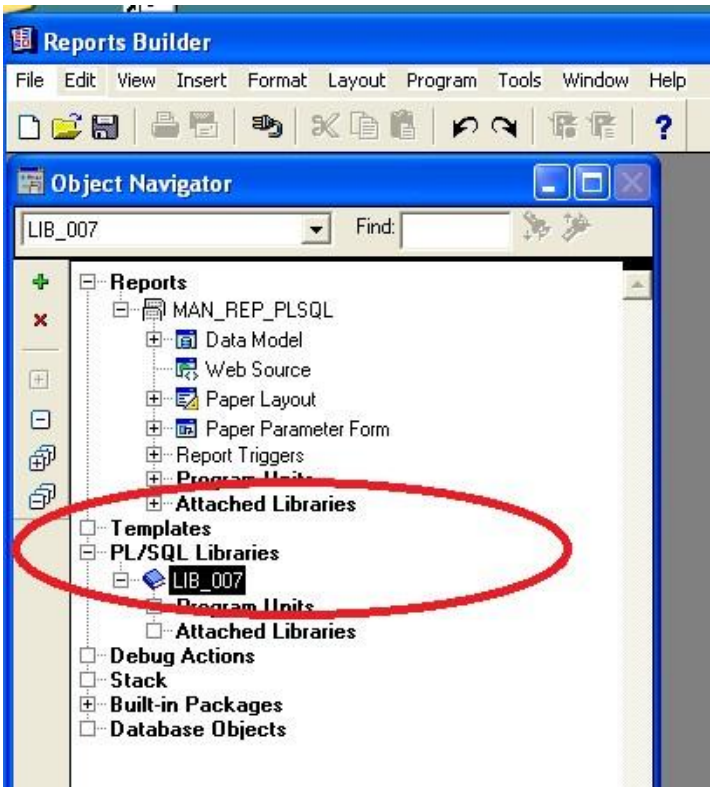
Step 1: Launch Reports Builder (or, if already open, choose File > New > Report).

Step 2: In the New-> Report then OK. Then select *"New Report Manually"* option and OK. Then system automatically bring it into blank Data Model screen.

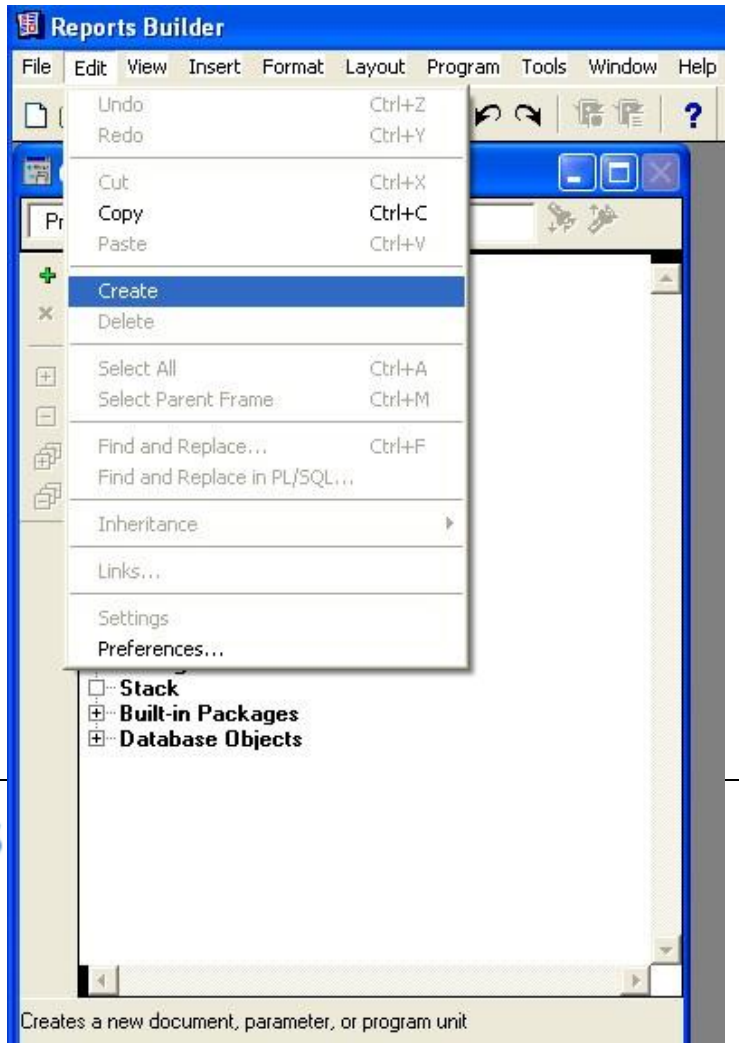


Step 3: Now For safer purpose first save the report as RDF extension, for this choose File > Save as option and then the system give a save opportunity and save the report as MAN_REP_PLSQL.RDF.

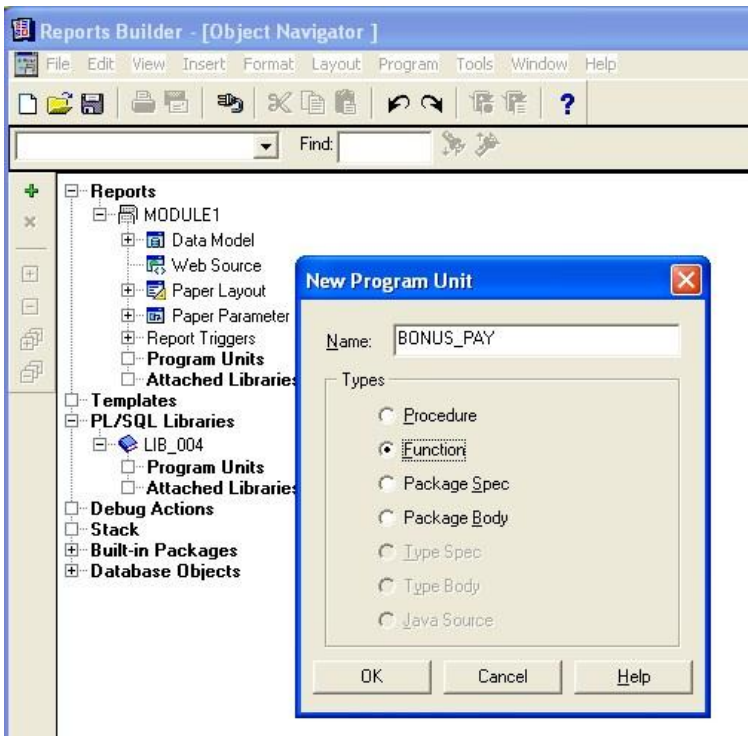
Step 4: Once again, on Object Navigator select PL/SQL Library, option then a new library opened. Which look as similar as below. The PL/SQL Library session is highlighted as *Red hallow circle*. The LIB_007 is default library name, if you need then rename it.



Step 5: Then Click the Program Units at that time the control placed under *"LIB_007"* library name. Now choose Edit > Create. Which look as below.



Step 6 : Once Create option is selected , then immediately the *New Program Unit* screen show , and on the name option enter BONUS_PAY as name , which look as below . The Type options are *Procedures, Functions and Packages*. This sample is example as function

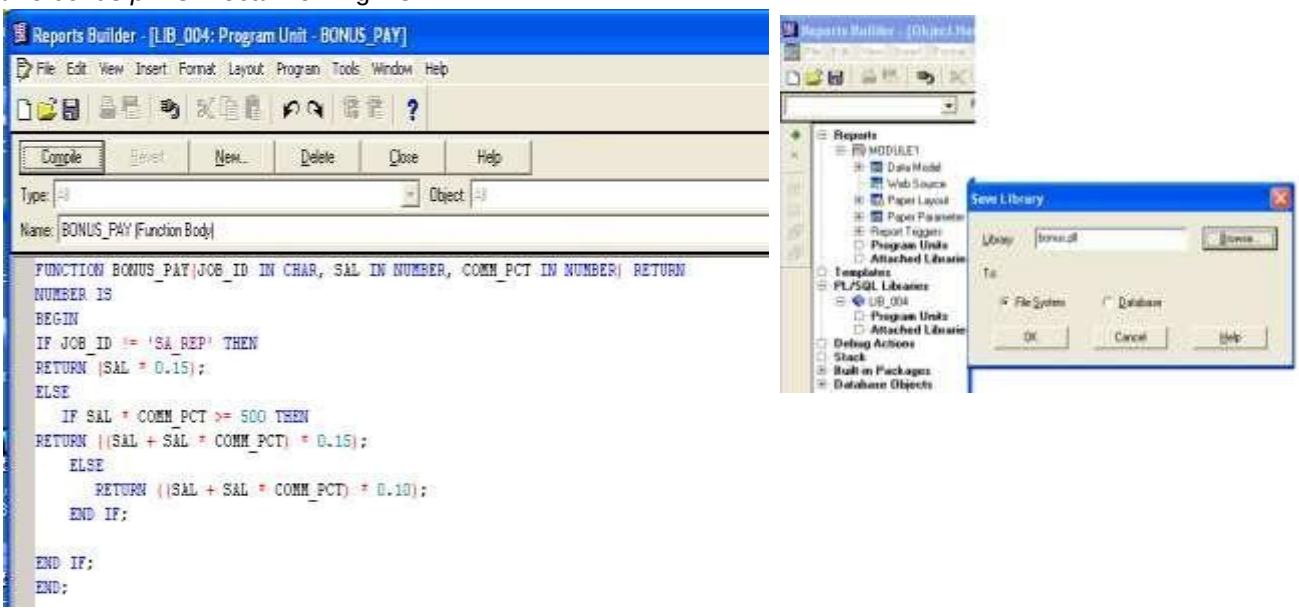


Step 7: Select Function, then click OK then system show the PL/SQL Editor then enter the following code as below.

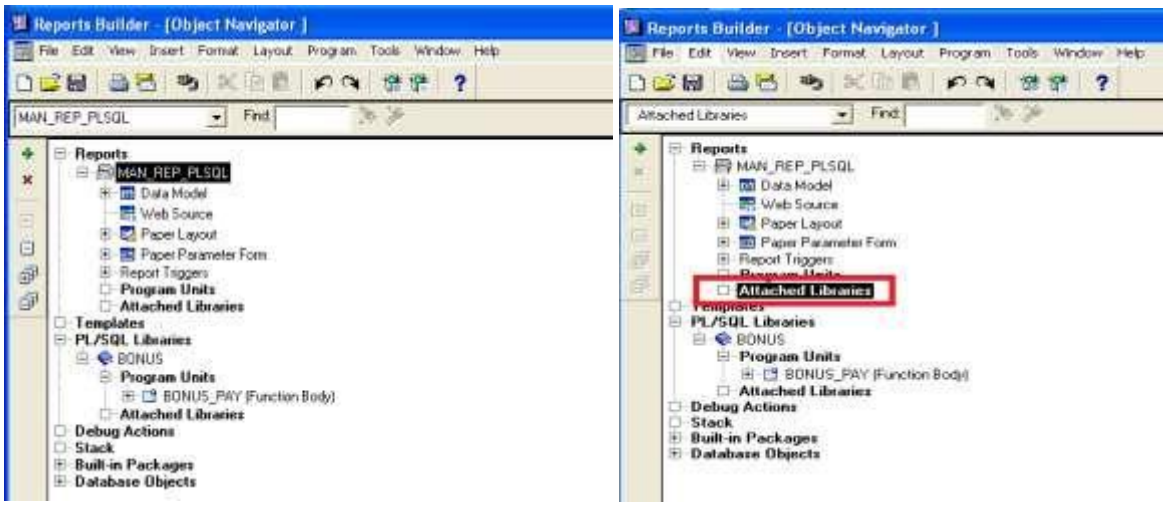
```
FUNCTION BONUS_PAY(JOB_ID IN CHAR, SAL IN NUMBER, COMM_PCT IN NUMBER) RETURN   NUMBER
IS

BEGIN

        IF JOB_ID != 'SA_REP' THEN

                RETURN (SAL * 0.15);
        ELSE

                IF SAL * COMM_PCT >= 500 THEN

                        RETURN ((SAL + SAL * COMM_PCT) * 0.15);
```

```
                ELSE
                        RETURN ((SAL + SAL * COMM_PCT) * 0.10);
                END IF;
        END IF;
END;
```

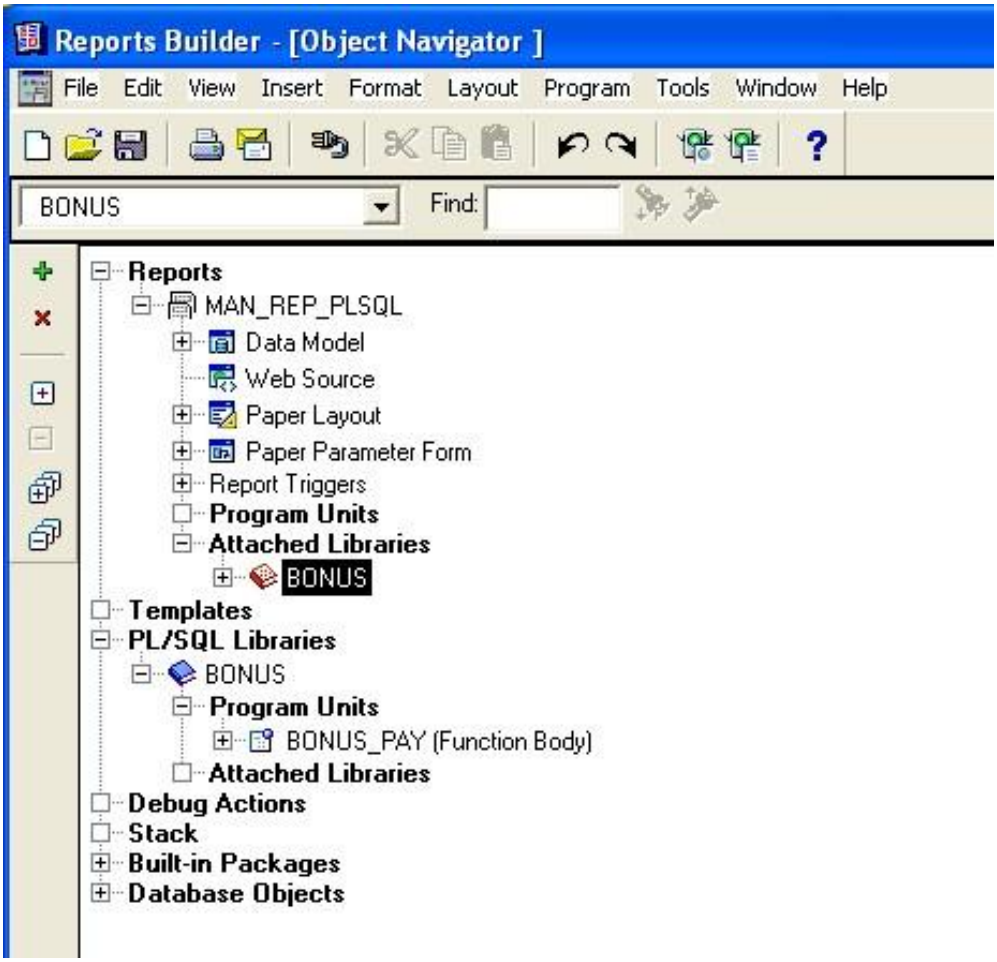
On the original editor screen as look below. If complete the coding then press Compile option, if there is no compilation error then press Close option. Finally Choose File > Save to save a PL/SQL Library in one OS file name *bonus.pll*. On local working PC



Step 8: Now come back to the MAN_REP_PLSQL report session on the Object Navigator. Now Attached Libraries option and Choose Edit > Create option and attach the *bonus.pll* file as created in the Library section. The steps look below,



Once attached the library file as in fine then the Object Navigator *Attached Libraries* section look something like this:

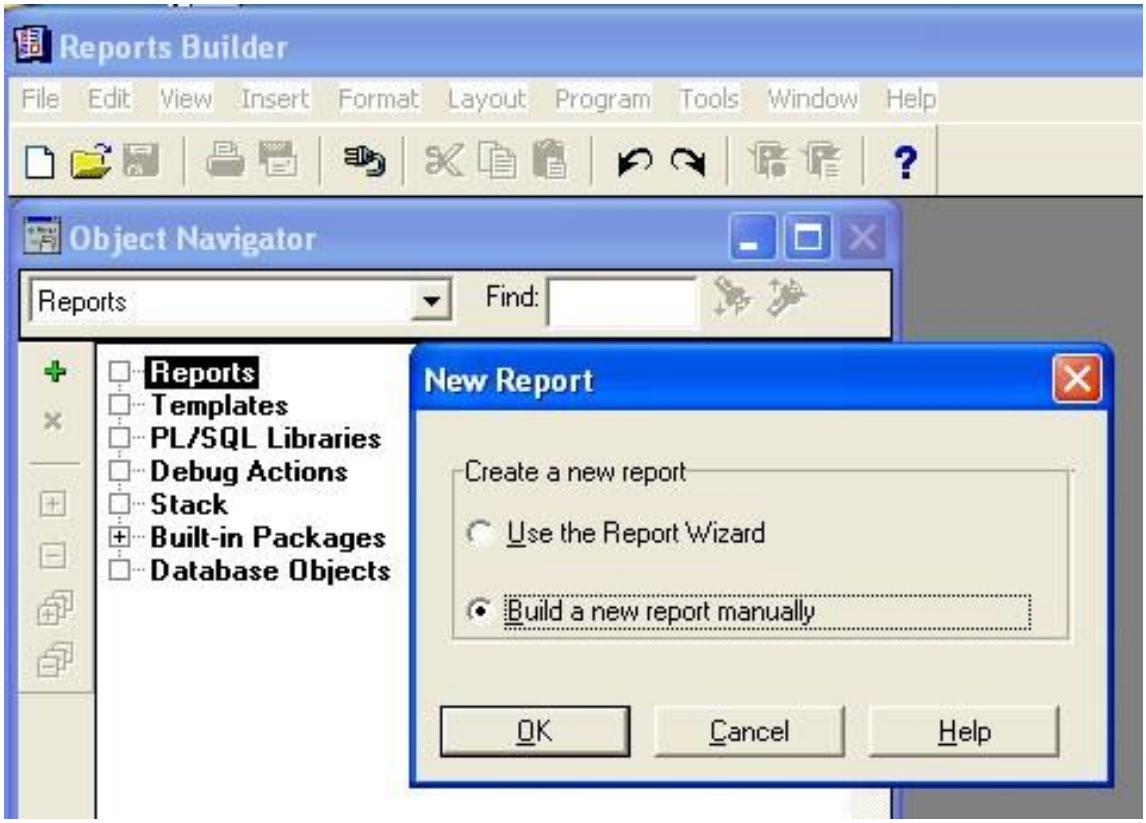


Like this way to all PL/SQL Libraries are attached. The second task is to add the library file in Attached *Libraries* section.

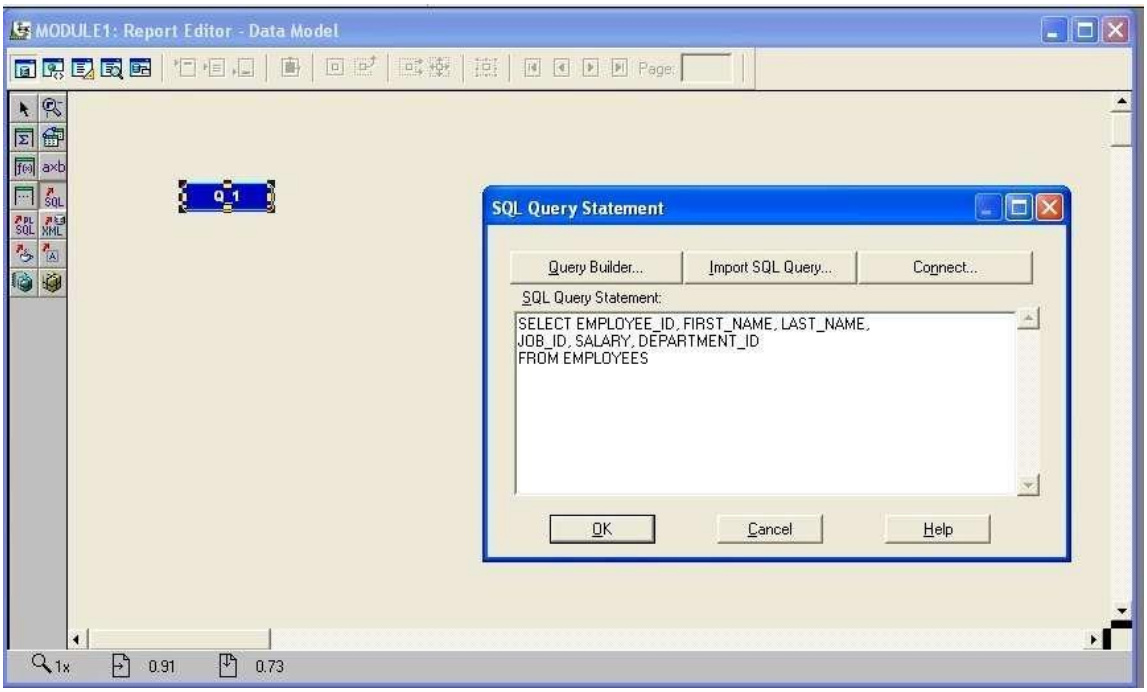
The better advice is to attached all libraries at the time of create new report and saved it as RPD file.

4. Program Units – Creating a formula column using PL/SQL library function

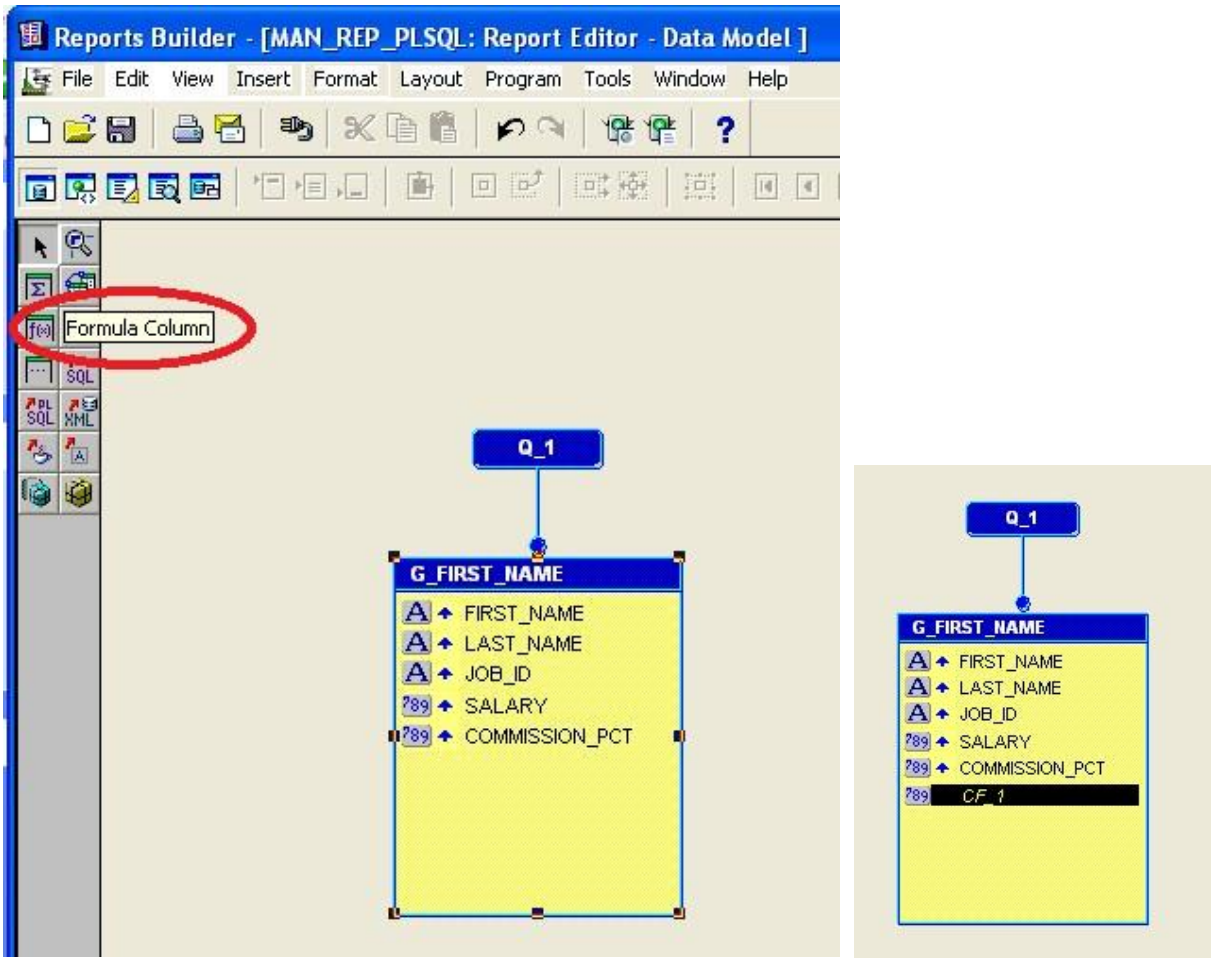
Step 1: In the New-> Report then OK. Then select *"New Report Manually"* option and OK. Then system automatically bring it into blank Data Model screen.



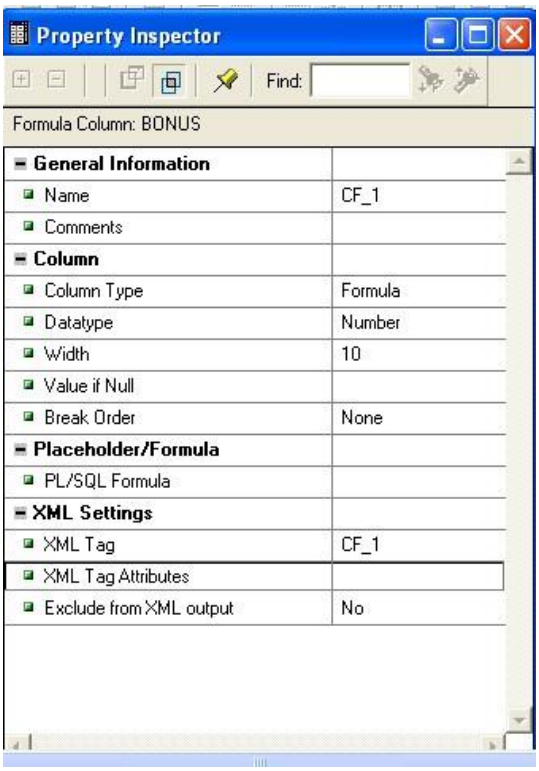
Step 2:Then.Click the SQL icon in the tool palette, then type the SQL as below.

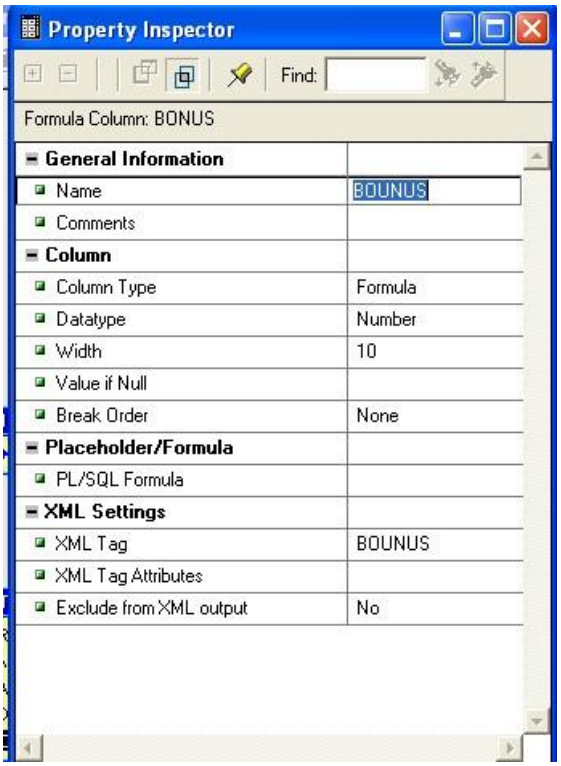


Step 3: .Click the Formula icon in the tool palette, then click in the G_FIRST_NAME group box, then system create a new create with default name CF_1 which is show in the adjacent picture.



Step 4: Double-click the new formula column object (CF_1) to display the Property plate and show as below first still. On name Property change it as BONUS and also change XML Tag value as same BONUS. It show on Second still. After this steps then go to PL/SQL Formula Property and click, then system show the PL/SQL Editor screen.



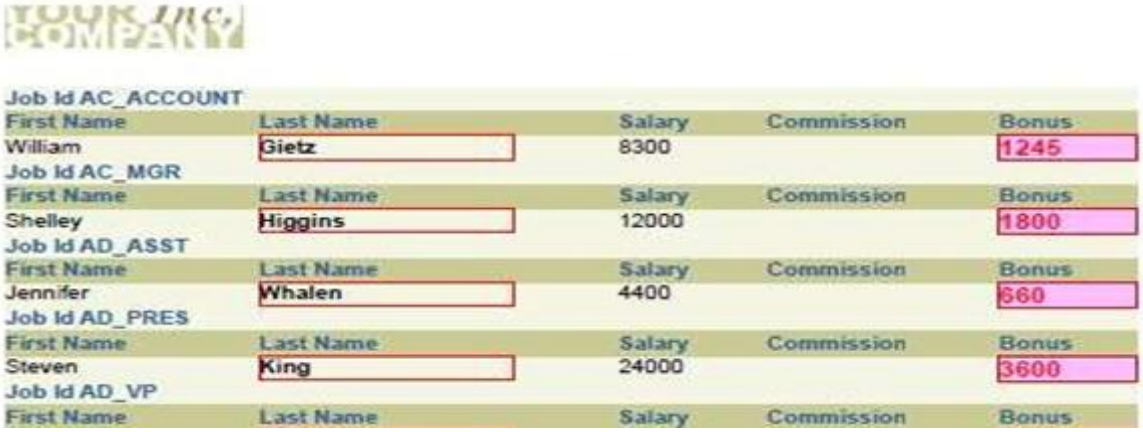


Step 5: The following code is add in the PL/SQL Editor screen.

```
Function BONUSFormula return Number is
Begin
        return BONUS_PAY(:JOB_ID, :SALARY, :COMMISSION_PCT);
End
```

The BONUS file has been attached to the report, which has the function called BONUS_PAY. This function has been called here to calculate the bonus.

The final report output will look like this (the calculated fields are marked in yellow)



Job Id AC_ACCOUNT				
First Name	Last Name	Salary	Commission	Bonus
William	Gietz	8300		1245
Job Id AC_MGR				
First Name	Last Name	Salary	Commission	Bonus
Shelley	Higgins	12000		1800
Job Id AD_ASST				
First Name	Last Name	Salary	Commission	Bonus
Jennifer	Whalen	4400		660
Job Id AD_PRES				
First Name	Last Name	Salary	Commission	Bonus
Steven	King	24000		3600
Job Id AD_VP				
First Name	Last Name	Salary	Commission	Bonus

Video5 Script: Vid6-Rep-Tools.avi (Screenplay of creating a Matrix report)

This video will show the screen capture of creating a report using manual data model.

Next Reading: "Oracle Reports Study Guide – Day3.pdf"