



ILP PROGRAM - ORACLE APPLICATIONS

Tata Consultancy Services

Oracle Reports Study Guide – Day3

Author: MD. Nazmul Hoda (TCS)
Creation Date: Jan 7, 2015
Last Updated: Jan 7, 2015
Document Ref: [ILP/ORACLEAPPS/REPORT/DAY3](#)
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

Document Control.....	2
How to use this manual	4
1. The Layout Editor	5
2. Program Units - Manual Data Model	7
3. Program Units - PL/SQL Libraries	15
4. Program Units – Creating a formula column using PL/SQL library function	21
5. Overview of Built in Packages	26
6. Triggers.....	29
7. Exercise 1: Client Master-Detail Report.....	31
8. Exercise 2: Bank Design.....	38
APPENDIX A: Table Creation Script	45
APPENDIX B: Table Creations Scripts for Bank Example	53
APPENDIX C: Further Reading	84

How to use this manual


This guide is continuation of “Oracle Reports Study Guide – Day2”. Continue reading it in the same way as Day1 and Day2 guide.

1. The Layout Editor

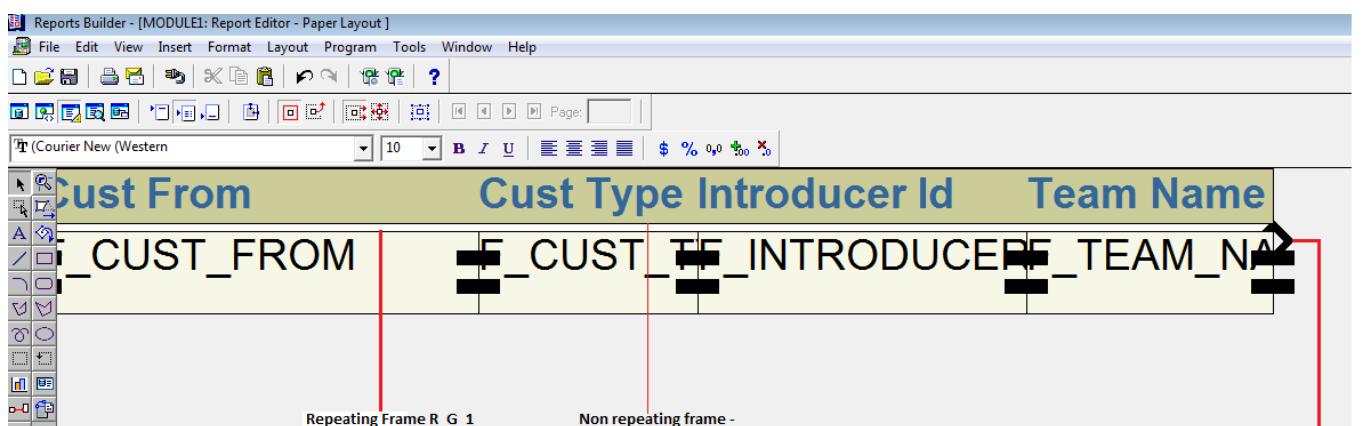


In this section you will learn about the structure of the frames and groups

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 the 3 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.



Repeating Frame R_G_1
(Holds repeating customer details, appears one per row)

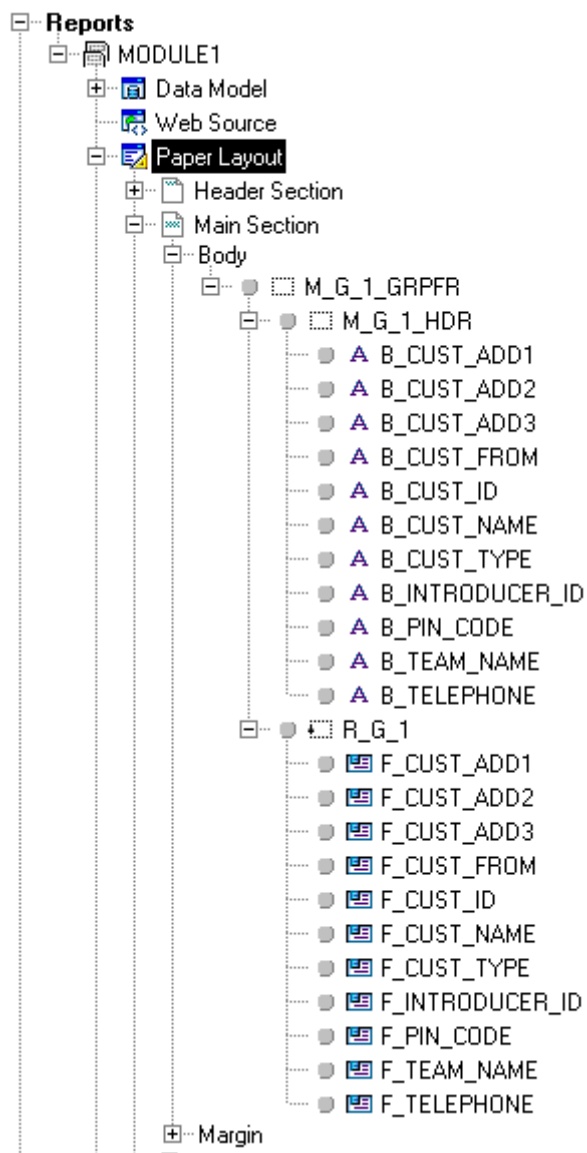
Non repeating frame - M_G_1_HDR
(Static frame - Holds Boilerplate text for headings)

Master Frame - M_G_1_GRPFR

Advanced Layout	
Print Object On	First Page
Base Printing On	Enclosing Object

Advanced Layout	
Print Object On	All Pages
Base Printing On	Anchoring Object

Advanced Layout	
Print Object On	First Page
Base Printing On	Enclosing Object



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

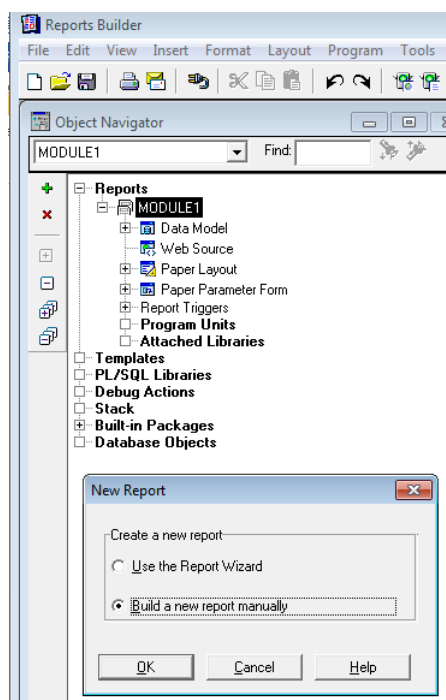
2. Program Units - Manual Data Model



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

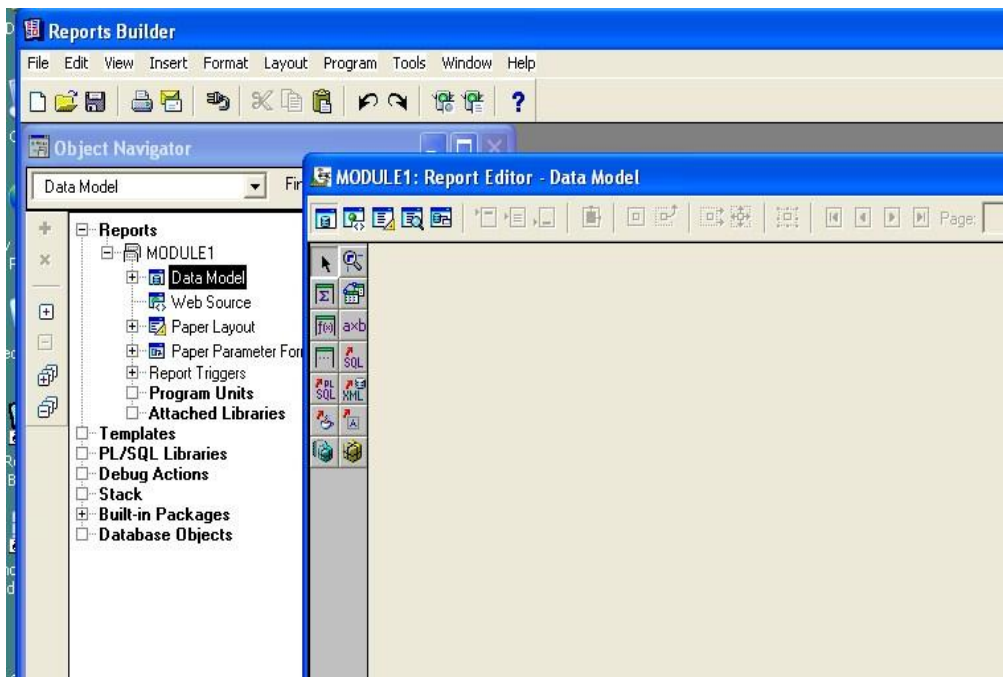
Open the Report Builder & Login as appsro user. We will be using a table – EMPLOYEE, if the table does not exist, use the script on in Appendix – A to create the table, and insert data into it.

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



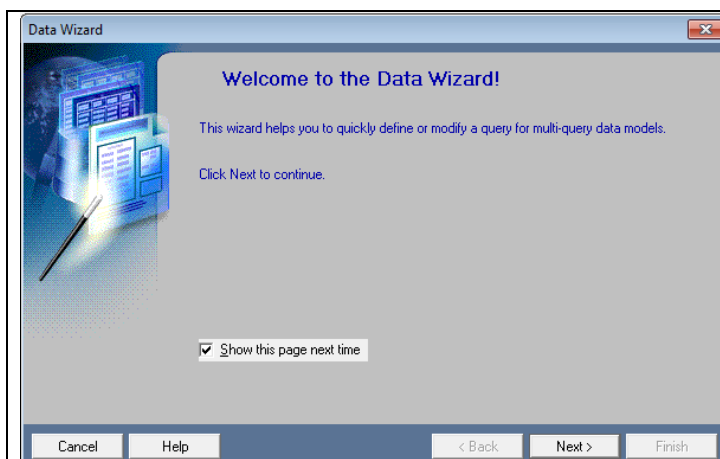
Open the report builder and click File > New from the toolbar, you will be prompted to select from 'Report Wizard' or 'Manual'.

Select 'Manual'

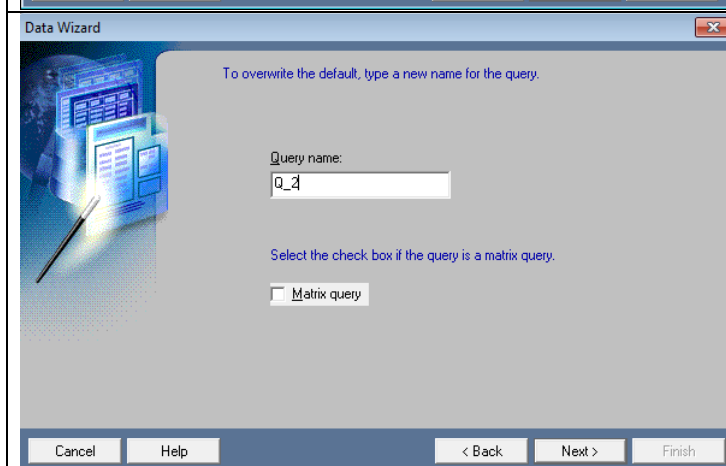


The System bring it into blank data model screen

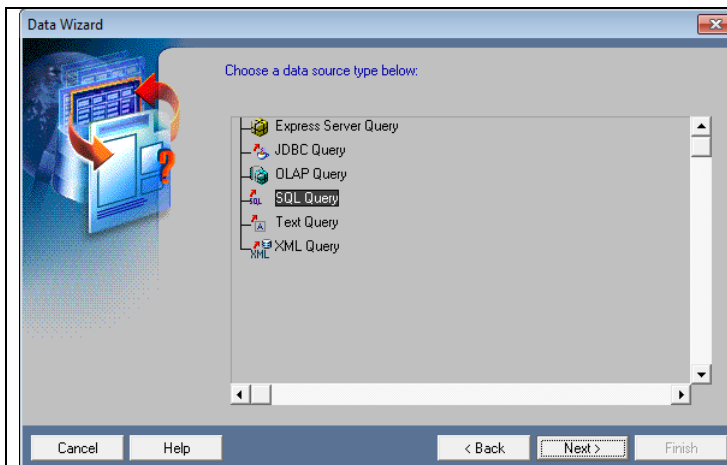
This is the full Blank Data Model Screen. As a programmatic approach, variety of tools available on left side. As this stage to double click on SQL icon and place on blank model session, then system automatically open the Query Builder tool



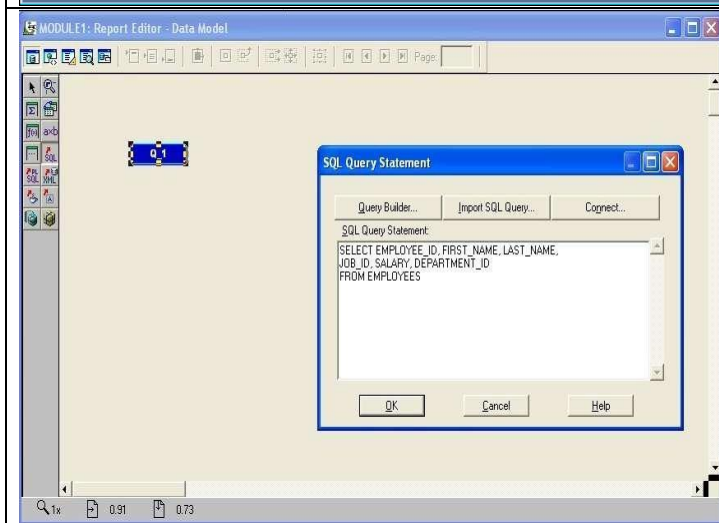
Right click and select Data Wizard, Select Next



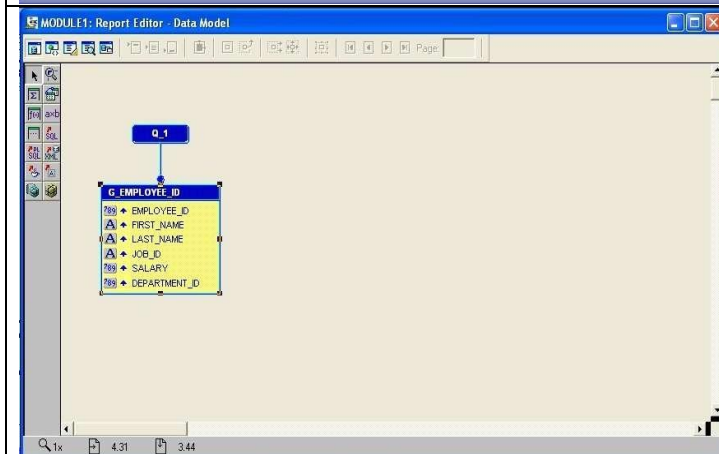
Supply the name of query, Select Next



Select 'SQL Query' as data source, click Next



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

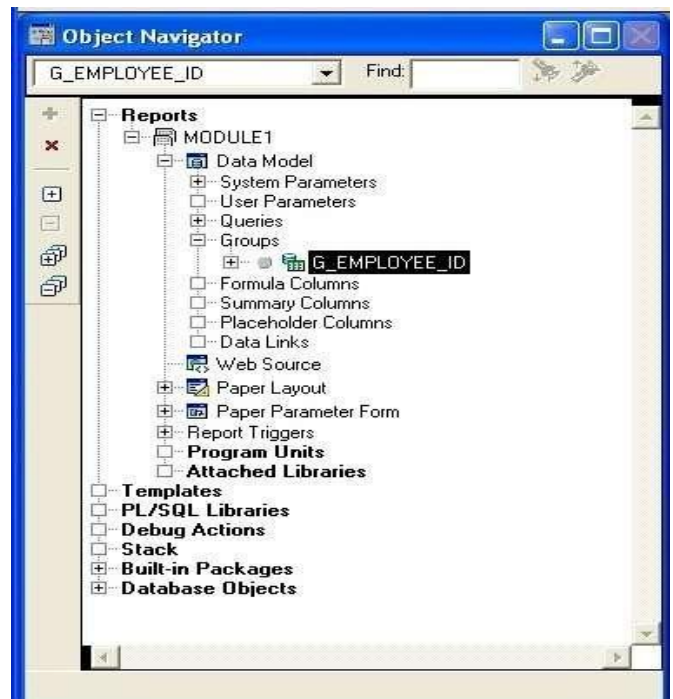


In the next screen, do not select any columns for 'Group Field', click Next.

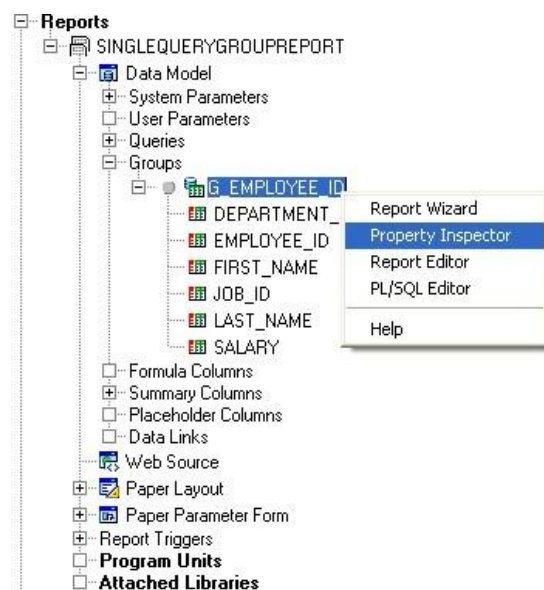
Also skip the 'calculate total' screen and hit 'Finish'.

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:

Reports Builder - [Property Inspector]

File Edit View Insert Format Layout Program Tools Window Help

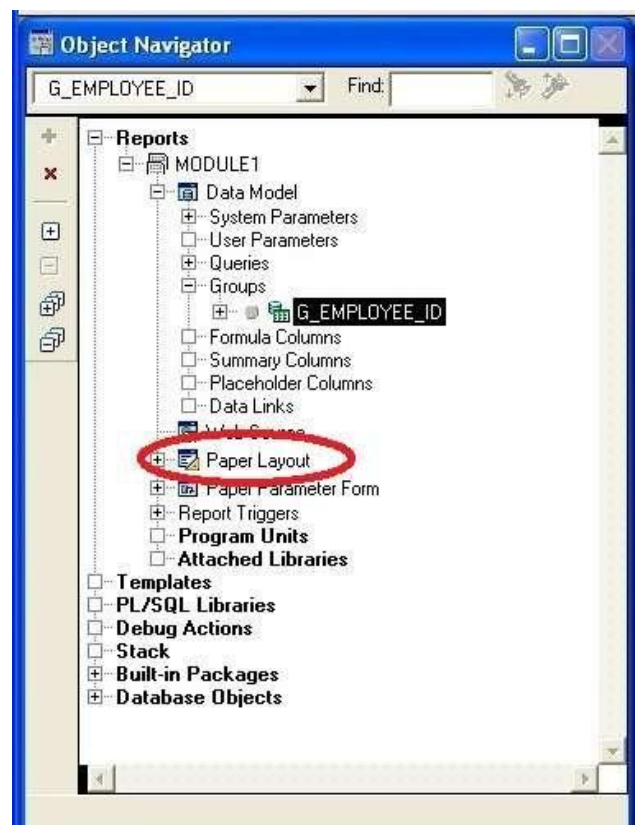
Find:

Group: G_EMPLOYEE_ID

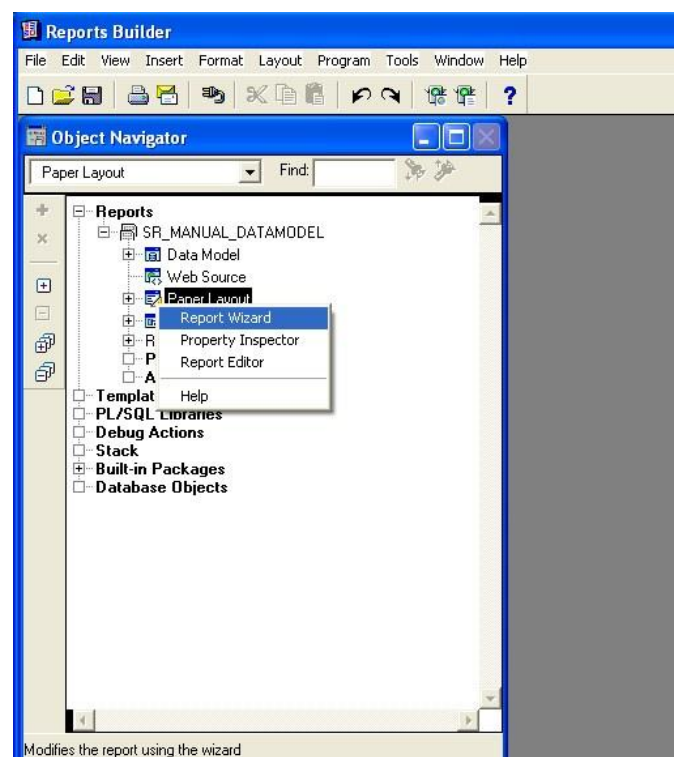
General Information	
Name	G_EMPLOYEE_ID
Comments	
Group	
Filter Type	None
XML Settings	
XML Tag	G_EMPLOYEE_ID
XML Tag Attributes	
Exclude from XML output	No
Outer XML Tag	LIST_G_EMPLOYEE_ID
Outer XML Attributes	

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”

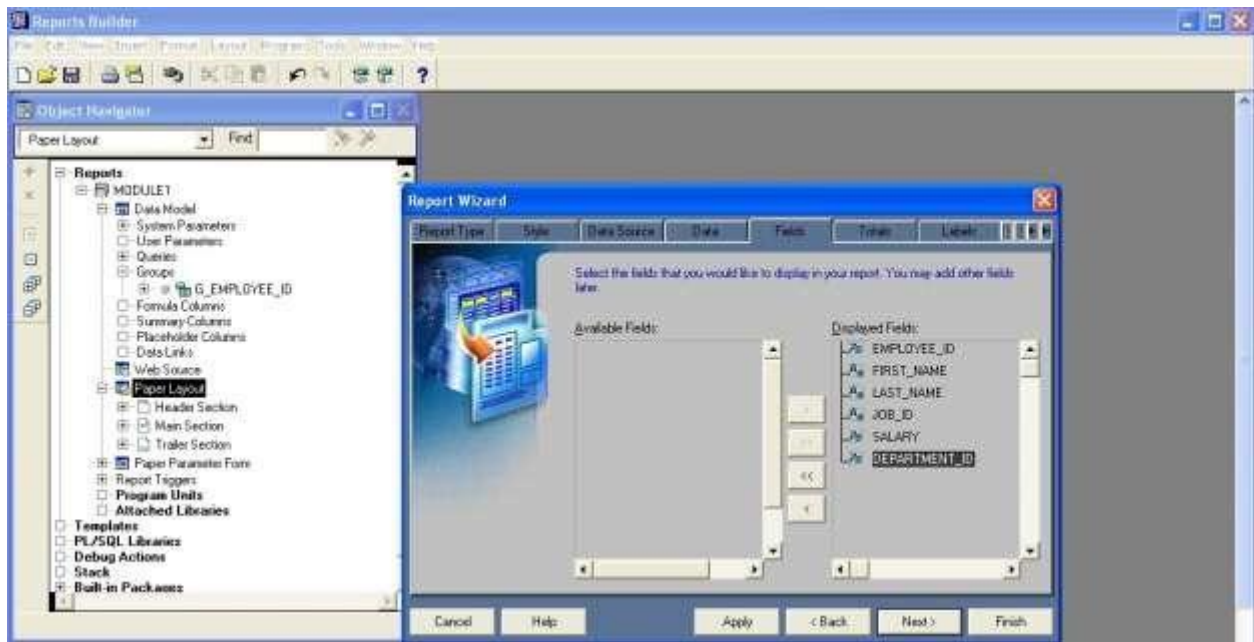
Now the Next task is to create a Layout using Manual option. For the purpose, place the mouse on Paper Layout then *Right Click* it, will show the selection option list



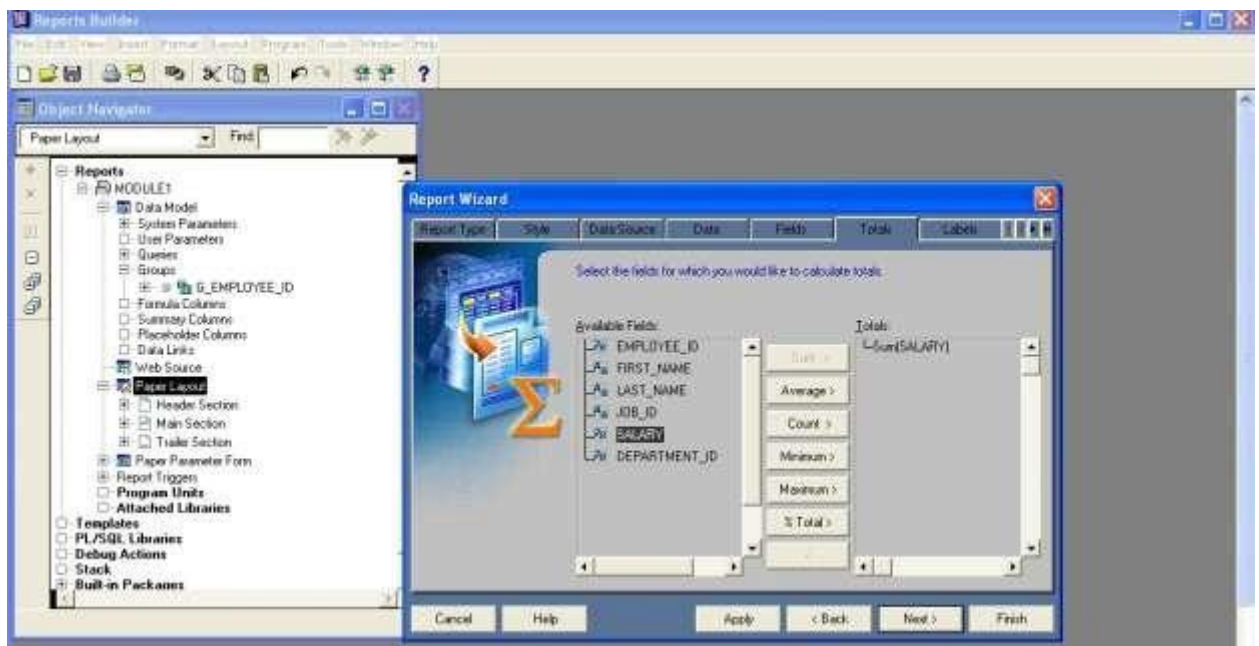
Then select the *Report wizard* Option and follow the wizard's Guide Flow path



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,

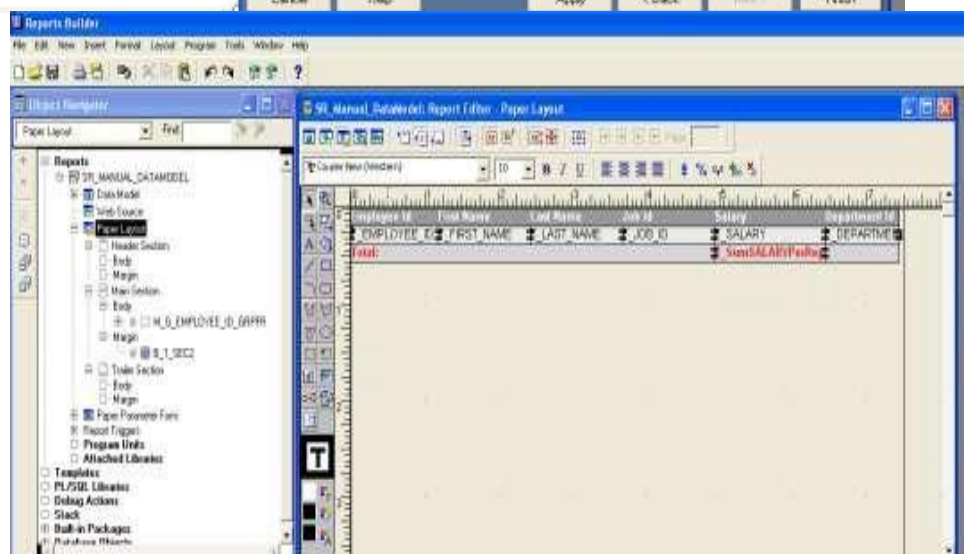


Finally select the corresponding Default Layout Templates

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.

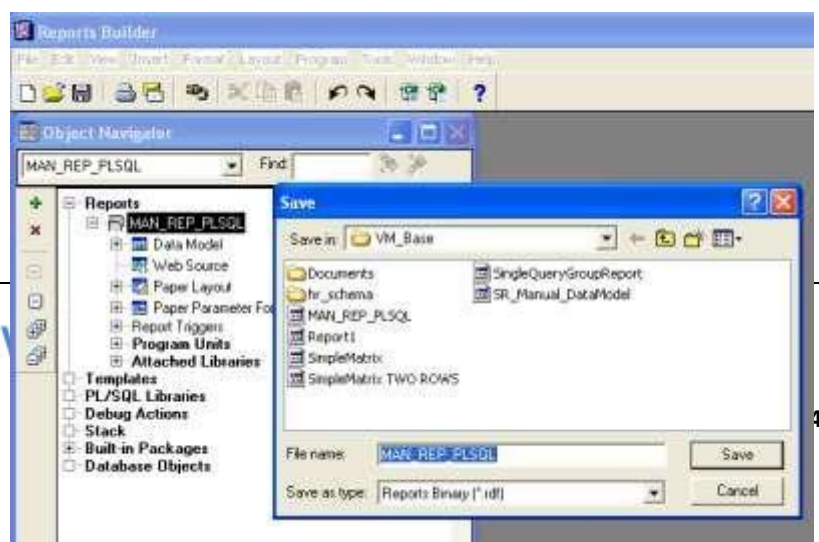
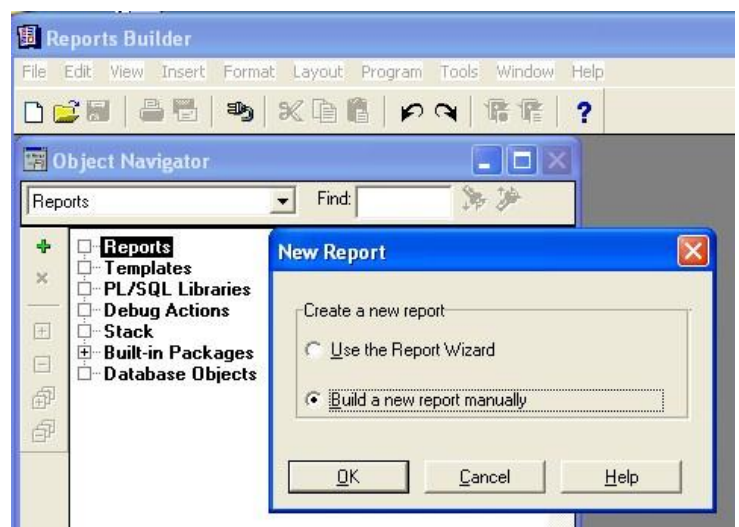
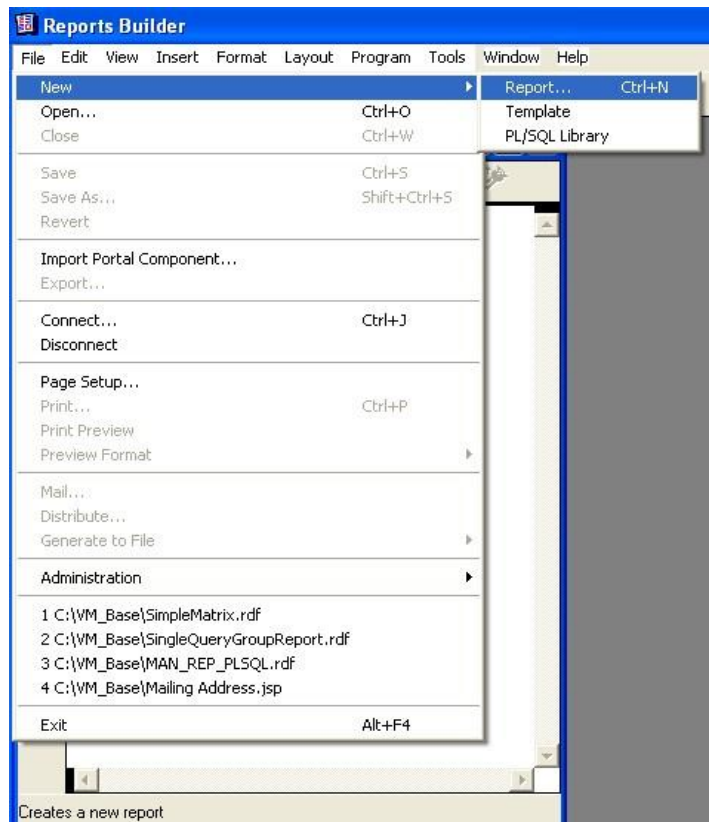
When we need to use a particular piece of code at multiple places, defining it in form of a PL/SQL library gives use several advantages:

- (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.

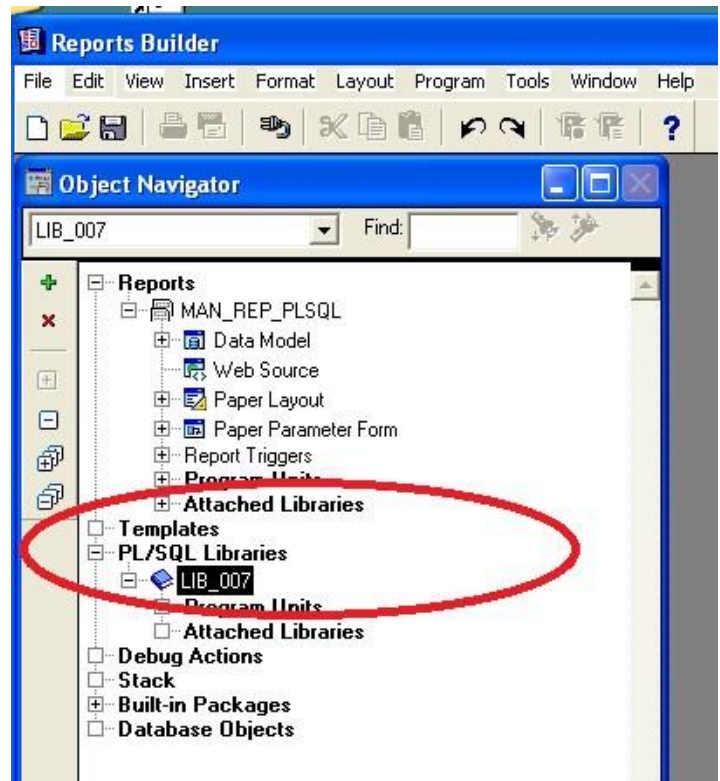
Step 1: Launch Reports Builder (or, if already open, choose File > New > Report).

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.

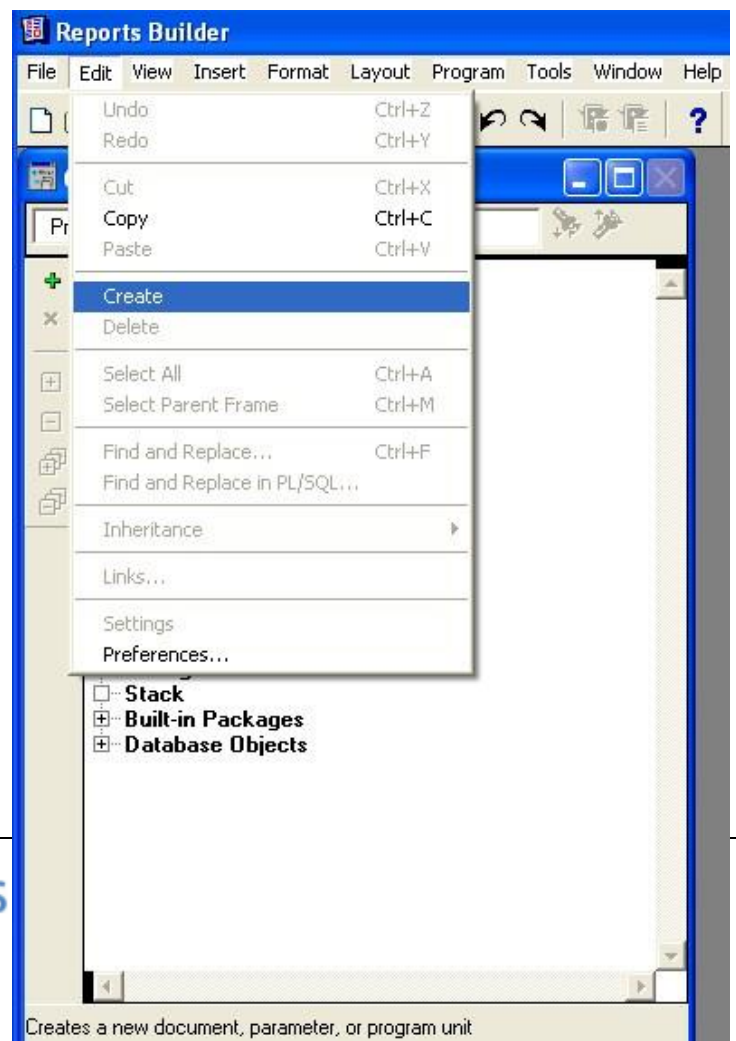


Step 3: Now For safer purpose first save the report as RDF extension, for this choose File > Save as option and then the system give a save opportunity and save the report as MAN_REP_PLSQL.RDF.

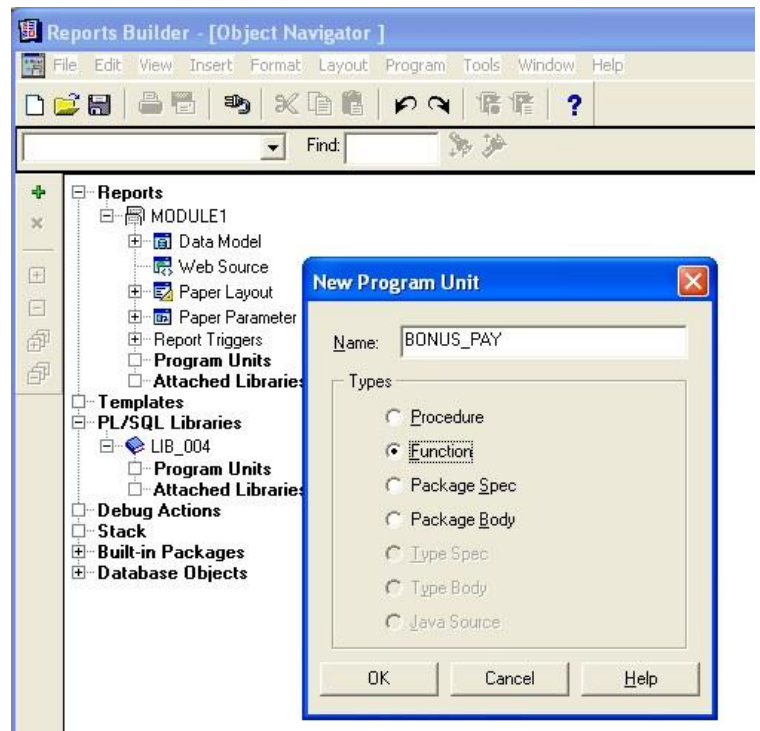
Step 4: Once again, on Object Navigator select PL/SQL Library, option then a new library opened. Which look as similar as below. The PL/SQL Library session is highlighted as *Red hallow circle*. The LIB_007 is default library name, if you need then rename it.



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

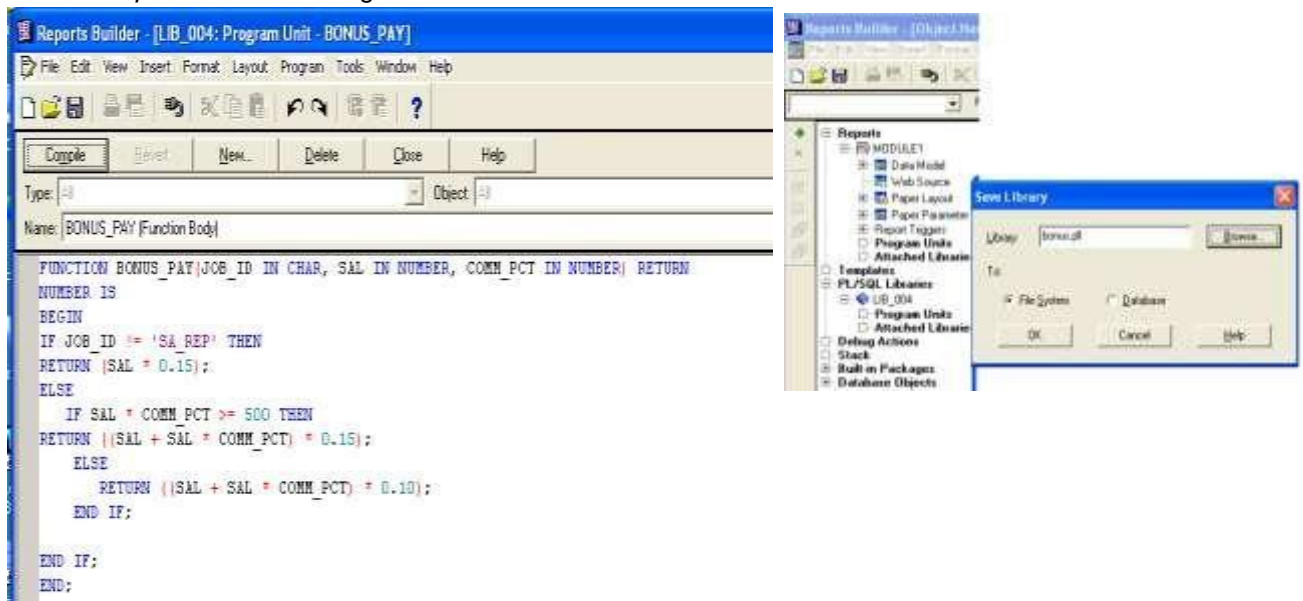


Step 7: Select Function, then click OK then system show the PL/SQL Editor then enter the following code as below.

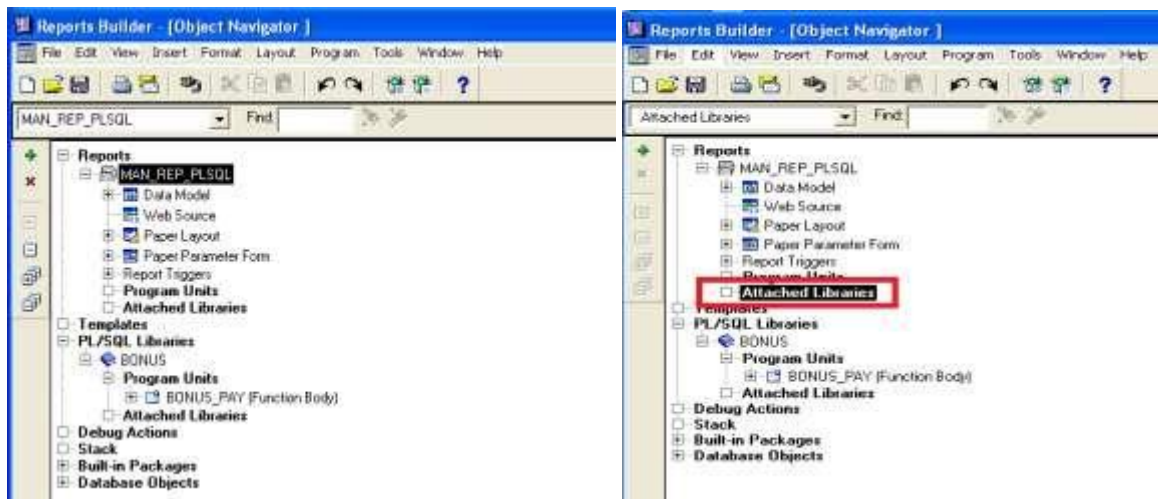
```
FUNCTION BONUS_PAY(JOB_ID IN CHAR, SAL IN NUMBER, COMM_PCT IN NUMBER) RETURN  NUMBER
IS
BEGIN
    IF JOB_ID != 'SA_REP' THEN
        RETURN (SAL * 0.15);
    ELSE
        IF SAL * COMM_PCT >= 500 THEN
            RETURN ((SAL + SAL * COMM_PCT) * 0.15);
        ELSE
            RETURN ((SAL + SAL * COMM_PCT) * 0.10);
        END IF;
    END IF;
END IF;
```

END;

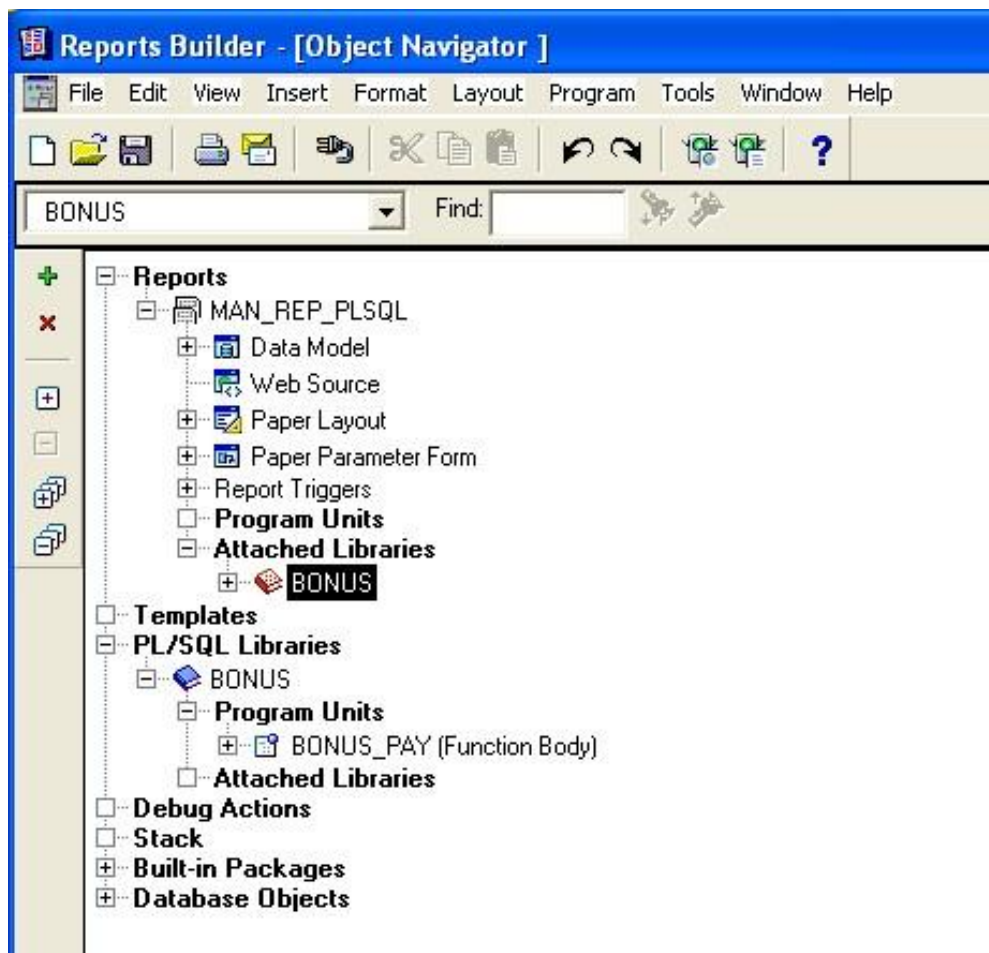
On the original editor screen as look below. If complete the coding then press Compile option, if there is no compilation error then press Close option. Finally Choose File > Save to save a PL/SQL Library in one OS file name *bonus.pll*. On local working PC



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:

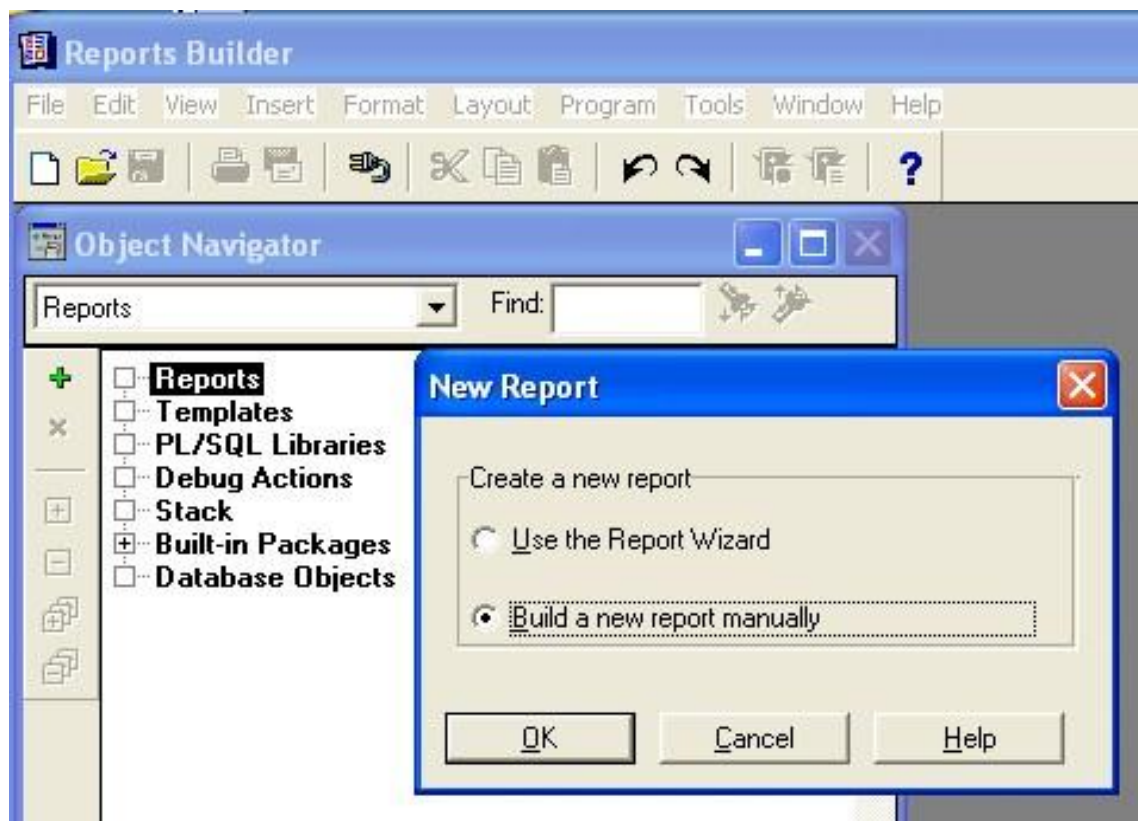


Like this way to all PL/SQL Libraries are attached. The second task is to add the library file in Attached_*Libraries* section.

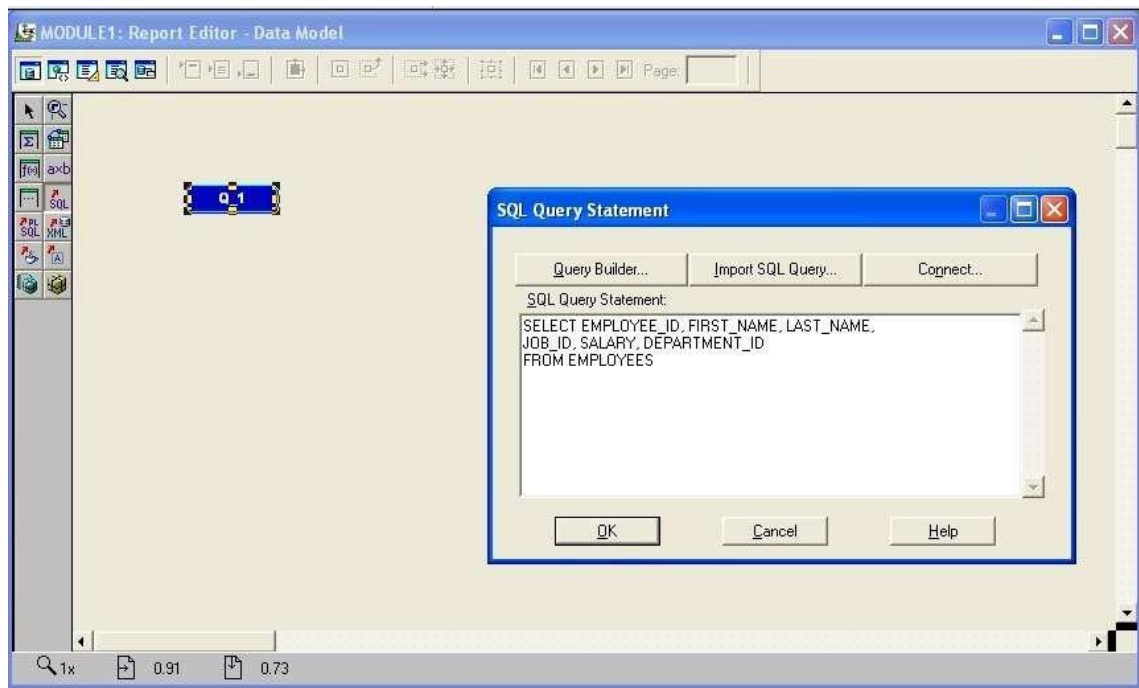
The better advice is to attached all libraries at the time of create new report and saved it as RPD file.

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

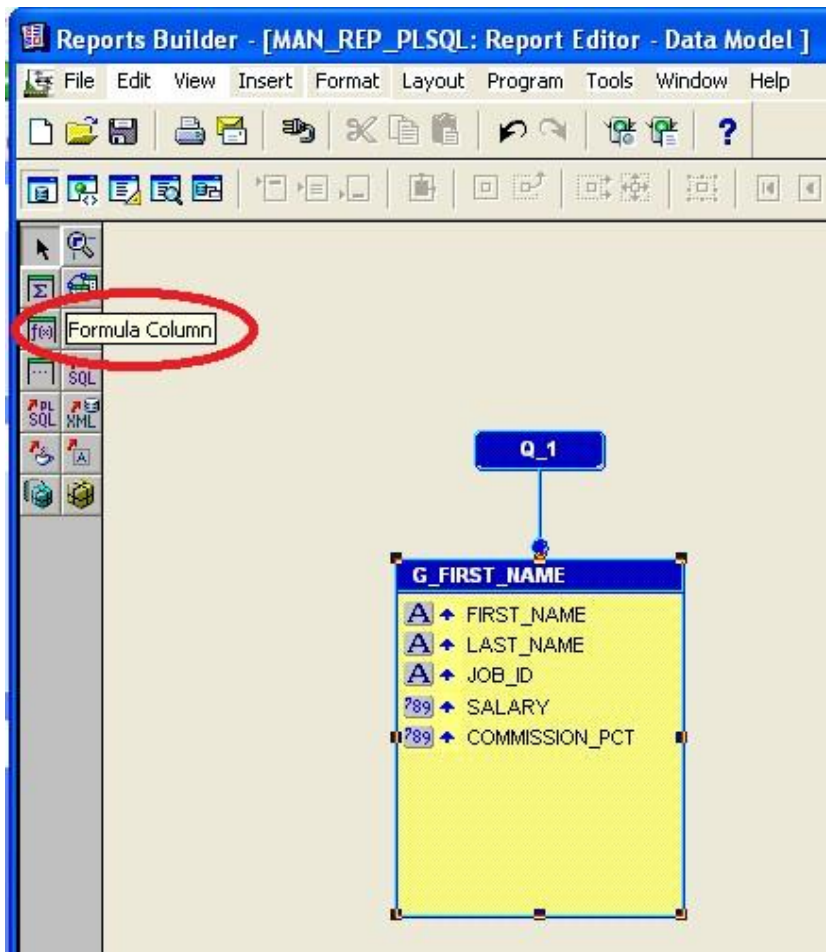
Step 1: In the New-> Report then OK. Then select “*New Report Manually*” option and OK. Then system automatically bring it into blank Data Model screen.



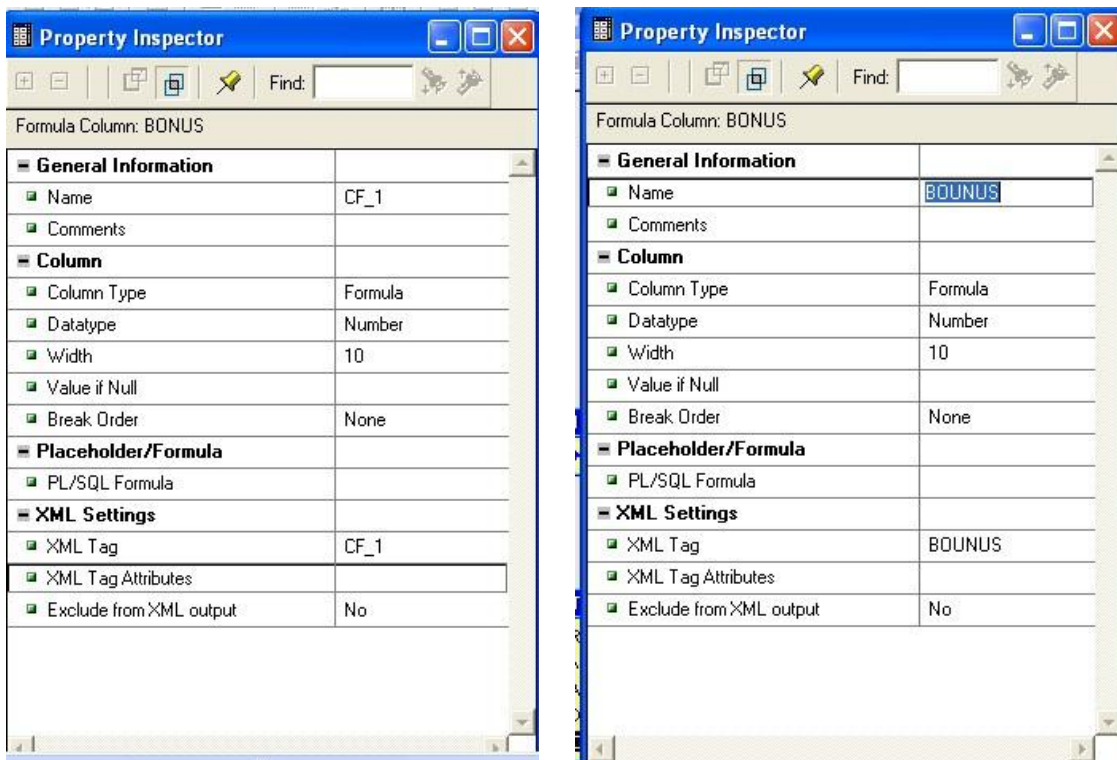
Step 2: Then, Click the SQL icon in the tool palette, then type the SQL as below.



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.

5. Overview of Built in Packages

Oracle Report Builder built-in package (SRW- Sql Report Writer)

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. You can reference the contents of the SRW package from any of your libraries or reports without having to attach it. However, you cannot reference its contents from within another product, for example, from SQL*Plus. Constructs found in a package are commonly referred to as "packaged" (that is, packaged functions, packaged procedures, and packaged exceptions).

Important SRW PACKAGES With sample

SRW.MESSAGE:

It is a Procedure that displays a message with the message number and text that you specify. It is mainly used to debug a report in Reports Builder.

SRW.MESSAGE(msg_number NUMBER, msg_text CHAR); Example:

```
function foo return boolean is
begin

if :sal <SRW.MESSAGE(100, 'Found a negative salary. Check the EMP table. '); 0
then

    raise SRW.PROGRAM_ABORT;

else

    :bonus := :sal * .01; if;

endreturn(true);
end;
```

SRW.PROGRAM ABORT:

This exception stops the report execution and raises the following error message: REP-1419: PL/SQL program aborted. SRW.PROGRAM_ABORT stops report execution when you raise it.

SRW.DO_SQL:

This procedure executes the specified SQL statement from within Reports Builder. The SQL statement can be DDL (statements that define data), or DML (statements that manipulate data). DML statements are usually faster when they are in PL/SQL, instead of in SRW.DO_SQL.

Since you cannot perform DDL statements in PL/SQL, the SRW.DO_SQL packaged procedure is especially useful for performing them within Reports Builder. Example:

```
FUNCTION CREATETABLE RETURN BOOLEAN IS
BEGIN

    SRW.DO_SQL('CREATE TABLE TEST_EMP (EMPNO NUMBER NOT NULL PRIMARY KEY, SAL
NUMBER (10,2)) PCTFREE 5 PCTUSED 75');

    RETURN (TRUE);

EXCEPTION
WHEN SRW.DO_SQL_FAILURE THEN

    SRW.MESSAGE(100, 'ERROR WHILE CREATING TEST_EMP TABLE.');
```

```
    RAISE SRW.PROGRAM_ABORT;

END;
```

SRW.DO_SQL_FAILURE:

Reports Builder raises this exception when the SRW.DO_SQL packaged procedure fails. This exception stops the report execution and raises the following error message:

REP-1425: Error running DO_SQL package – REP-msg ORA-msg.

SRW.GET_REPORT_NAME:

This function returns the file name of the report being executed.

SRW.GET_REPORT_NAME (report_name); Example:

```
AfterPForm return boolean is
function    my_var varchar2(80);
BEGIN
    SRW.GET_REPORT_NAME (my_var);
    SRW.MESSAGE(0, 'Report Filename = ' || my_var);

    RETURN (TRUE); 7    END;
```

SRW.RUN_REPORT:

This procedure synchronously executes the specified report within the context of the currently running report.

SRW.RUN_REPORT ("report=test.rdf ... ") [SRW.SET_FIELD](#):

This procedure sets the value of a character, number, or date field. This is useful when you want to conditionally change a field's value. SRW.SET_FIELD (object_id, text CHAR | number NUM | date DATE); Example:

Suppose you want to conditionally change the number of a field, based on each employee's salary. In the format trigger for the field, you could type the following:

```
FUNCTION CHGFIELD RETURN BOOLEAN IS
    TMP NUMBER;

BEGIN
    if :sal >= 5000
    then
        tmp := :sal * 1.10;
        srw.set_field (0, tmp);
    else
        srw.set_field (0, 4000);
    end if;

    end

    RETURN (TRUE);
END;
```

SRW.SET_FIELD should be used only to change the contents of a field's data type, not change the field to a different data type.

Some other familiar Packages are

- [SRW.SET_FONT_FACE](#): This procedure specifies font face for a CHAR, DATE, or NUMBER field.
SRW.SET_FONT_FACE('arial');
- [SRW.SET_FONT_SIZE](#): This procedure specifies font size for a CHAR, DATE, or NUMBER field.
SRW.SET_FONT_SIZE(10);
- [SRW.SET_FONT_STYLE](#): This procedure specifies font style for a CHAR, DATE, or NUMBER field.
SRW.SET_FONT_STYLE(SRW.ITALIC_STYLE);
- [SRW.SET_FORMAT_MASK](#): This procedure specifies the format mask for the DATE or NUMBER field.
SRW.SET_FORMAT_MASK('mask');
- [SRW.SET_TEXT_COLOR](#): This procedure specifies the global text color of the CHAR, DATE, or NUMBER field.
SRW.SET_TEXT_COLOR('color');

6. Triggers



In this section you will learn how to use different triggers to perform functions based on certain events

Triggers is a piece of code, which is executed when a particular event occurs. For example a **Before Parameter Form** trigger will fire just before a parameter form is invoked. There is limited set of triggers available inside oracle reports, which are discussed in this section:

Field Level Trigger:

If you need to hide sequence no just return (false) in format trigger. See below

If you have SALARY named field in report layout. Just click the field and press F11 functional key format trigger will open. In that format trigger already in return (true). You change the Following code

```
if Bonus > 5000 then
return (true); else
return (false); end if;
```

Apart from the field level trigger,

Before Parameter Form

Fires before the Runtime Parameter Form are displayed. Can access the PL/SQL global Variables, report level columns and manipulate accordingly.

After Parameter Form

Fires after the Runtime Parameter form are displayed. Used to validate the parameter values.

Before Report

Fires before the report is executed but after the queries is parsed and data is fetched.

Between Pages

Fires before each page of the report are formatted, except the very first page. This page is used to customize page formatting. After Report fires after the report previewer are exited, or after report output is sent to a specified destination

One sample After Parameter Form trigger:

In the Object Navigator, under your report name, expand the Report Triggers node. Right-click AFTER PARAMETER FORM, then choose PL/SQL Editor from the pop-up menu. In the PL/SQL Editor, use the template to enter the following code:

```
function AfterPForm      return boolean is begin
    srw.set_format_order(srw.main_section,
srw.header_section,      srw.trailer_section);
    return (TRUE);
end;
```

**Video6: Script:** Summary of built in packages

In this section, we have learnt about the built in packages. These packages are used to extract system level information or take actions which cannot be done otherwise. For Example we can use SRW.DO_SQL package to execute a DDL statement at runtime. Also functions like SRW.SET_FONT_STYLE can be used to change the FONT on the fly. SRW.MESSAGE function can be used to pop custom error message at runtime.

We have also learnt about the triggers which can be used to take a particular action at runtime. For example between pages trigger can be used to display page level summary information.

7. Exercise 1: Client Master-Detail Report



In this exercise, you will develop a master-detail form. The table creation and population script has been provided

Specification:

Print the Client's Personal Details with all Contact Numbers. The Report contain Only Client details with none of the Heading details. Create a report with First Name Middle Name, Last name, Stage, Pin code, and Phone Numbers. The Phone Numbers supposed to print all numbers with one time the personal Details

Table definition:

CLNT_MSTR		CLNT_DTLS	
Name	Null? Type	Name	Null? Type
<hr/>		<hr/>	
CLNTID	NOT NULL NUMBER(5)	CLNTID	NOT NULL NUMBER(5)
FNAME	NOT NULL VARCHAR2(25)	CONTACTTYPE	NOT NULL CHAR(1)
MNAME	VARCHAR2(25)	CONTACTNO	NOT NULL VARCHAR2(15)
LNAME	NOT NULL VARCHAR2(25)		
ADDR1	NOT NULL VARCHAR2(40)		
ADDR2	NOT NULL VARCHAR2(40)		
CITY	NOT NULL VARCHAR2(25)		
PINCODE	NOT NULL VARCHAR2(6)		
DOB	NOT NULL DATE		
PANCOPY	NOT NULL CHAR(1)		
VERIFIEDDATA	NOT NULL CHAR(1)		

Report Output:

The Cleint Personal DETAILS with All Contact Numbers

Alex	Austin	Fernandes	MUMBAI	400016	
<div>ContacttypeContactno</div> <div>O6480903</div>					<div>Bonus</div> <div>1245</div>
Anil	Arun.	Dhone	MUMBAI	400016	
<div>ContacttypeContactno</div> <div>R4372247</div>					<div>Bonus</div> <div>1800</div>
Ashwini	Dilip	Joshi	MUMBAI	400016	
					<div>Bonus</div> <div>660</div>
					<div>Bonus</div> <div>3600</div>

Table creation and data population script:

Table Creation	<pre> DROP TABLE clnt_dtls; DROP TABLE clnt_mstr; COMMIT; CREATE TABLE"CLNT_MSTR" ("CLNTID" NUMBER(5) NOT NULL, "FNAME" VARCHAR2(25) NOT NULL, "MNAME" VARCHAR2(25), "LNAME" VARCHAR2(25) NOT NULL, "ADDR1" VARCHAR2(40) NOT NULL, "ADDR2" VARCHAR2(40) NOT NULL, "CITY" VARCHAR2(25) NOT NULL, "PINCODE" VARCHAR2(6) NOT NULL, "DOB" DATE NOT NULL, "PANCOPY" CHAR(1) NOT NULL, "VERIFIEDDATA" CHAR(1) NOT NULL, CONSTRAINT "PK_CLNTMSTR_CLNTID" PRIMARY KEY("CLNTID")); CREATE TABLE"CLNT_DTLS" ("CLNTID" NUMBER(5) NOT NULL, "CONTACTTYPE" CHAR(1) NOT NULL, "CONTACTNO" VARCHAR2(15) NOT NULL, CONSTRAINT "FK_CLNTDTLS_CLNTID" FOREIGN KEY("CLNTID") REFERENCES"CLNT_MSTR"("CLNTID")); </pre>
----------------	---

<p>Insert data for CLNT_MSTR Table</p>	<pre> INSERT INTO"CLNT_MSTR" (CLNTID, FNAME, MNAME, LNAME, ADDR1, ADDR2, CITY, PINCODE, DOB, PANCOPY, VERIFIEDDATA) VALUES (1, 'Ivan', 'Nelson', 'Bayross', 'F- 12, Diamond Palace, West Avenue,', 'North Avenue, Santacruz (West)', 'MUMBAI', '400056', '25-Jun-52', 'Y', 'Y'); INSERT INTO"CLNT_MSTR" (CLNTID, FNAME, MNAME, LNAME, ADDR1, ADDR2, CITY, PINCODE, DOB, PANCOPY, VERIFIEDDATA) VALUES (2, 'Chriselle', 'Ivan', 'Bayross', 'F-12, Diamond Palace, West Avenue,', 'North Avenue, Santacruz (West)', 'MUMBAI', '400056', '29- Oct-82', 'N', 'Y'); INSERT INTO"CLNT_MSTR" (CLNTID, FNAME, MNAME, LNAME, ADDR1, ADDR2, CITY, PINCODE, DOB, PANCOPY, VERIFIEDDATA) VALUES (3, 'Mamta', 'Arvind', 'Muzumdar', 'Magesh Prasad', 'Saraswati Baug, Jogeshwari(E)', 'MUMBAI', '400060', '28-Aug-75', 'Y', 'Y'); INSERT INTO"CLNT_MSTR" (CLNTID, FNAME, MNAME, LNAME, ADDR1, ADDR2, CITY, PINCODE, DOB, PANCOPY, VERIFIEDDATA) VALUES (4, 'Chhaya', 'Sudhakar', 'Bankar', '4, Sampada', 'Kataria Road, Mahim', 'MUMBAI', '400016', '06-Oct-76', 'Y', 'Y'); </pre>
--	--

```

INSERT INTO"CLNT_MSTR" (CLNTID, FNAME, MNAME,
LNAME, ADDR1, ADDR2, CITY, PINCODE, DOB, PANCOPY,
VERIFIEDDATA)
VALUES (5, 'Ashwini',
'Dilip', 'Joshi',
'104, Vikram Apts. Bhagat Lane', 'Shivaji
Park,Mahim', 'MUMBAI', '400016', '20-Nov-78', 'Y',
'Y');

INSERT INTO"CLNT_MSTR" (CLNTID, FNAME, MNAME,
LNAME, ADDR1, ADDR2, CITY, PINCODE, DOB, PANCOPY,
VERIFIEDDATA)
VALUES (6, 'Hansel',
'I.', 'Colaco', '12,
Radha kunj, N. C. Kelkar Road', 'Dadar', 'MUMBAI',
'400016', '1-Jan-82', 'N', 'Y');

INSERT INTO"CLNT_MSTR" (CLNTID, FNAME, MNAME,
LNAME, ADDR1, ADDR2, CITY, PINCODE, DOB, PANCOPY,
VERIFIEDDATA)
VALUES (7,
'Anil', 'Arun.', 'Dhone', 'A/14,
Shanti Society, Mogal Lane,', 'Mahim', 'MUMBAI',
'400016', '12-Oct-83', 'N', 'Y');

INSERT INTO"CLNT_MSTR" (CLNTID, FNAME, MNAME,
LNAME, ADDR1, ADDR2, CITY, PINCODE, DOB, PANCOPY,
VERIFIEDDATA)
VALUES (8, 'Alex',
'Austin', 'Fernandes',
'5, Vagdevi, Senapati Bapat Rd.', 'Dadar',
'MUMBAI', '400016', '30-Sep-62', 'Y', 'Y');
INSERT INTO"CLNT_MSTR" (CLNTID, FNAME, MNAME,
LNAME, ADDR1, ADDR2, CITY, PINCODE, DOB, PANCOPY,
VERIFIEDDATA)
VALUES (9, 'Ashwini',
'Shankar', 'Apte',
'A-10 Nutan Vaishali', 'Shivaji Park, Mahim',
'MUMBAI', '400016', '19-Apr-79', 'Y', 'Y');
INSERT INTO"CLNT_MSTR" (CLNTID, FNAME, MNAME,
LNAME, ADDR1, ADDR2, CITY, PINCODE, DOB, PANCOPY,
VERIFIEDDATA)

```

	<pre>VALUES (10, 'Namita', 'S.', 'Kanade', 'B10, Makarand Society', 'Cadal Road, Mahim', 'MUMBAI', '400016', '10-Jun-78', 'Y', 'Y'); COMMIT;</pre>
--	--

<p>Insert Data for CLNT_DTLS Table</p>	<pre> INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (1, 'R', '6405853'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (1, 'O', '6134553'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (1, 'M', '9820178955'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (1, 'O', '6134571'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (2, 'R', '60455754'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (2, 'O', '6134571'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (3, 'R', '8324567'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (3, 'O', '6197654'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (4, 'R', '4449852'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (4, 'O', '8741370'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (5, 'R', '4302934'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (5, 'O', '2819964'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (6, 'R', '4217592'); </pre>
--	--

<pre>INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (7, 'R', '4372247'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (8, 'O', '6480903'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (9, 'R', '4313408'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (9, 'M', '9821176651'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (10, 'R', '4362680'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (10, 'O', '8973355'); INSERT INTO"CLNT_DTLS" (CLNTID, CONTACTTYPE, CONTACTNO) VALUES (10, 'M', '9820484648'); COMMIT;</pre>
--

Hint:

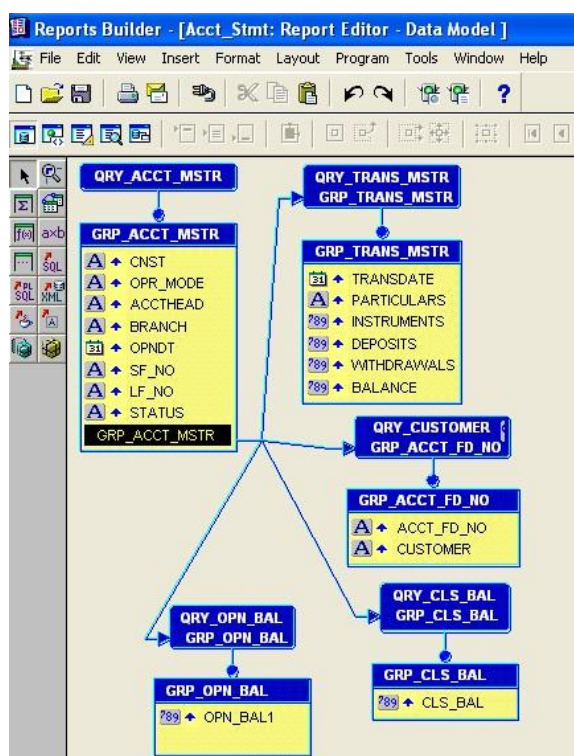
Use the below Query as data source:

```
SELECT ALL CLNT_MSTR.FNAME,
CLNT_MSTR.MNAME,
CLNT_MSTR.LNAME,
CLNT_MSTR.CITY,
CLNT_MSTR.PINCODE,
CLNT_DTLS.CONTACTTYPE,
CLNT_DTLS.CONTACTNO
```

```
FROM CLNT_DTLS, CLNT_MSTR
WHERE (CLNT_DTLS.CLNTID = CLNT_MSTR.CLNTID)
```

8. Exercise 2: Bank Design

This report is based on data stored inside 5 tables. These table's data is interrelated as shown in the below data model. In the below model, you can see 5 queries (blue bubbles) and 5 groups. The detail of these queries are given below. Use these queries to create the data source of the queries. The table creation script is available in Appendix B.



1. QRY_ACCT_MSTR :

```
SELECT (CASE WHEN ACCT_MSTR.TITLE is null THEN '<Non-Corporate Account>'
          WHEN ACCT_MSTR.TITLE is not null THEN ACCT_MSTR.TITLE END) AS ACCTHEAD,
       (SELECT BRANCH_MSTR.NAME FROM BRANCH_MSTR
        WHERE BRANCH_MSTR.BRANCH_NO = ACCT_MSTR.BRANCH_NO) AS BRANCH,
       (SELECT DISTINCT SPRT_DOC.TYPE FROM SPRT_DOC
        WHERE SPRT_DOC.ACCT_CODE = ACCT_MSTR.CUR_ACCT_TYPE) AS CNST,
       (CASE WHEN ACCT_MSTR.OPR_MODE = 'SI' THEN 'Singly'
             WHEN ACCT_MSTR.OPR_MODE = 'AS' THEN 'Anyone or Survivor'
             WHEN ACCT_MSTR.OPR_MODE = 'JO' THEN 'Jointly'
             WHEN ACCT_MSTR.OPR_MODE = 'ES' THEN 'Everyone or Survivor' END) AS OPR_MODE,
       ACCT_MSTR.OPNDT, ACCT_MSTR.SF_NO, ACCT_MSTR.LF_NO,
       (CASE WHEN ACCT_MSTR.STATUS = 'A' THEN 'Active'
```

```

        WHEN ACCT_MSTR.STATUS = 'C' THEN 'Cancelled'
        WHEN ACCT_MSTR.STATUS = 'M' THEN 'Matured' END) As STATUS FROM ACCT_MSTR WHERE
ACCT_MSTR.ACCT_NO LIKE :PRM_ACCTNO.

```

2. QRY_TRANS_MS

```

SELECT TRANS_MSTR.DT AS TRANSDATE,
      ((CASE WHEN TRANS_MSTR.DR_CR = 'D' THEN 'By '
        WHEN TRANS_MSTR.DR_CR = 'W' THEN 'To ' END) ||
(CASE WHEN TRANS_MSTR.TYPE = 'C' THEN TRANS_MSTR.PARTICULAR
        WHEN NOT (TRANS_MSTR.TYPE = 'C') THEN
(SELECT TRANS_DTLS.PAYTO || ': CLG' FROM TRANS_DTLS
        WHERE TRANS_DTLS.TRANS_NO = TRANS_MSTR.TRANS_NO) END)) AS PARTICULARS,
(CASE WHEN NOT (TRANS_MSTR.TYPE = 'C') THEN
      (SELECT TRANS_DTLS.INST_NO FROM TRANS_DTLS
        WHERE TRANS_DTLS.TRANS_NO = TRANS_MSTR.TRANS_NO) END) AS INSTRUMENTS,
(CASE WHEN TRANS_MSTR.DR_CR = 'D' THEN TRANS_MSTR.AMT END) AS DEPOSITS,
(CASE WHEN TRANS_MSTR.DR_CR = 'W' THEN TRANS_MSTR.AMT END) AS WITHDRAWALS,
      TRANS_MSTR.BALANCE
FROM TRANS_MSTR WHERE TRANS_MSTR.ACCT_NO LIKE :PRM_ACCTNO
      AND TRANS_MSTR.DT BETWEEN :PRM_STRTDT AND :PRM_ENDDT

```

3. QRY_CUSTOMER

```

SELECT ACCT_FD_NO, (SELECT CUST_MSTR.CUST_NO || ' - ' ||
      CUST_MSTR.FNAME || ' ' || CUST_MSTR.MNAME || ' ' ||
      CUST_MSTR.LNAME || ' - ' || CUST_MSTR.DOB_INC
FROM CUST_MSTR WHERE CUST_MSTR.CUST_NO = ACCT_FD_CUST_DTLS.CUST_NO) AS
      Customer FROM ACCT_FD_CUST_DTLS

```

```

WHERE ACCT_FD_CUST_DTLS.ACCT_FD_No
      LIKE :PRM_ACCTNO

```

4. QRY_CLS_BAL

```

SELECT TRANS_MSTR.BALANCE AS CLS_BAL
FROM TRANS_MSTR WHERE RowID =(SELECT MAX(RowID)
      FROM TRANS_MSTR
      WHERE ACCT_NO = :PRM_ACCTNO AND
      TRANS_MSTR.DT < (TO_DATE(:PRM_ENDDT) + 1))

```

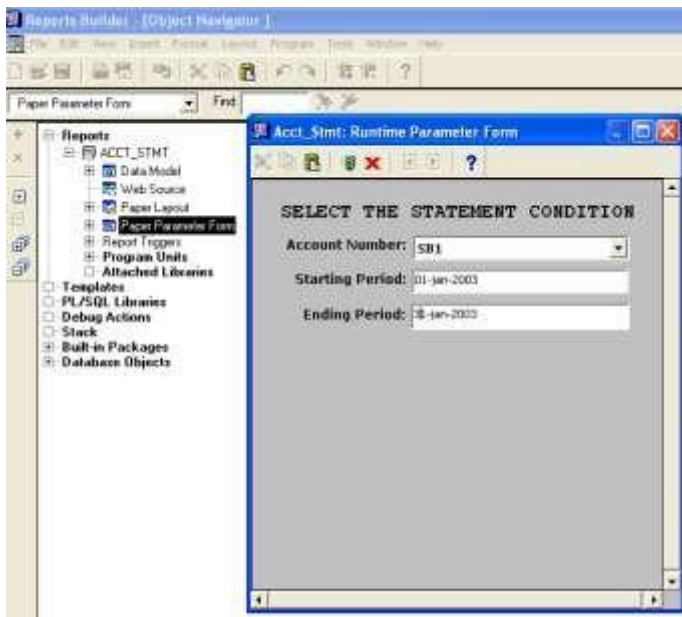
5. QRY_OPN_BAL

```

SELECT TRANS_MSTR.BALANCE AS OPN_BAL
FROM TRANS_MSTR WHERE RowID =(SELECT MAX(RowID)
      FROM TRANS_MSTR
      WHERE ACCT_NO = :PRM_ACCTNO AND TRANS_MSTR.DT < :PRM_STRTDT)

```

The report should pop-up a parameter form at run time as shown below:



Develop three reports:

A. Bank Design – Report 1: Statement of Accounts report:

The report output should look like this:

Statement Of Account																													
<div style="border: 1px solid black; padding: 5px;"> Account Information: <div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> Account No.: <input type="text" value="SB1"/> Branch: <input type="text" value="Vile Parle (HO)"/> </div> <div style="margin-bottom: 5px;"> Account Name: <input type="text" value="<Non-Corporate Account>"/> </div> <div style="margin-bottom: 5px;"> Constitution: <input type="text" value="Individuals / Savings Bank Account"/> </div> <div style="display: flex; justify-content: space-between;"> Opening Date: <input type="text" value="05-JAN-03"/> Operated: <input type="text" value="Singly"/> </div> </div>																													
<div style="border: 1px solid black; padding: 5px;"> Internal Information: <div style="display: flex; justify-content: space-between; margin-top: 5px;"> S. F. Number: <input type="text" value="SF-0001"/> Ledger Folio: <input type="text" value="JAN03-05"/> </div> </div>																													
<div style="border: 1px solid black; padding: 5px;"> Account Holder's Information: <div style="margin-top: 5px;"> Name: <input type="text" value="C1 - Ivan Nelson Bayross - 25-JUN-52"/> </div> </div>																													
<div style="border: 1px solid black; padding: 5px; display: flex; justify-content: space-between; margin-top: 5px;"> From Date: <input type="text" value="01-JAN-03"/> To Date: <input type="text" value="31-JAN-03"/> </div>																													
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Date</th> <th style="width: 30%;">Particulars</th> <th style="width: 15%;">Instruments</th> <th style="width: 15%;">Deposits</th> <th style="width: 15%;">Withdrawals</th> <th style="width: 15%;">Balance Dr/Cr</th> </tr> </thead> <tbody> <tr> <td>01-JAN-03</td> <td>By Opening Bal b/d</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>05-JAN-03</td> <td>By Initial Payment</td> <td></td> <td style="text-align: right;">500</td> <td></td> <td style="text-align: right;">500</td> </tr> <tr> <td>31-JAN-03</td> <td>To Closing Bal c/d</td> <td></td> <td></td> <td></td> <td style="text-align: right;">500</td> </tr> </tbody> </table>						Date	Particulars	Instruments	Deposits	Withdrawals	Balance Dr/Cr	01-JAN-03	By Opening Bal b/d					05-JAN-03	By Initial Payment		500		500	31-JAN-03	To Closing Bal c/d				500
Date	Particulars	Instruments	Deposits	Withdrawals	Balance Dr/Cr																								
01-JAN-03	By Opening Bal b/d																												
05-JAN-03	By Initial Payment		500		500																								
31-JAN-03	To Closing Bal c/d				500																								
Page					1																								

B. Bank Design – Report2 : Customer Information Report

The report output should look like this:

Customer's Information	
Customer No.: C4	Date Of Birth: 06-OCT-76
Cust. Name: Chhaya Sudhakar Bankar	
PAN No. Submitted: Yes	Photograph At: D:/ClntPht/C4.gif
Form 60 Submitted: Yes	Signature At: D:/ClntSgnt/C4.gif
Customer's Address Information:	
Address Type: Current Residential Address	
Address: 4, Sampada, Kataria Road, Mahim,	
City: Mumbai	State: Maharashtra Pincode: 400016
Customer's Contact Information:	
Contact Type	Contact Details
Residence	24449852
Office	28741370
Customer's Bank Account Information:	
Account No.: SB3 Account Type: Savings Bank	
Acct. Head: <Non-Corporate Account>	
Constitution: Individuals / Savings Bank Accou	Opened On: 22-JAN-03
Operated By: Singly	Status: Active
Branch: B1 - Vile Parle (HO)	Cur. Balance: 500
Account No.: CA4 Account Type: Current Account	
Acct. Head: Sun's Pvt. Ltd.	
Constitution: Limited Companies	Opened On: 05-FEB-03
Operated By: Anyone Or Survivor	Status: Active
Branch: B1 - Vile Parle (HO)	Cur. Balance: 2000

C. Bank Design – Report 3: Fixed Deposit Report

The report output should look like this:

Fixed Deposit Information	
Branch:	B1 - Vile Parle (HO)
F. D. Head:	Puru Hsg. Soc
Constitution:	Clubs / Societies
Assurer:	Existing Account: Current Account - CA7
Bank's Internal Information:	
S. F. No.:	SF-0013
Verified By:	Peter Iyer Joseph - Clerk
Officer's Signature Present:	<input checked="" type="checkbox"/> Yes
Manager's Signature Present:	<input checked="" type="checkbox"/> Yes
Corporate Customer's Address Information:	
Address Type:	Head Office
Address:	G-2, Puru Hsg. Society, Senapati Bapat Rd., Dadar,
City:	Mumbai
State:	Maharashtra
Pincode:	400016
Corporate Customer's Contact Information:	
Contact Type	Contact Details
Telephone	24301090
Telephone	24301196
Fixed Deposit Holder's Information:	
Name:	Hansel I. Colaco - 01-JAN-82
Name:	Alex Austin Fernandes - 30-SEP-62
Fixed Deposit Details:	
F. D. No.:	F5
F. D. Type:	Standard
Issued On:	24-JUN-03
Period:	183 Days
Deposit:	2000
Due On:	24-DEC-03
Int. Rate:	6
Maturity:	2060.16
F. D. Status:	Active
F. D. Auto Renewal:	<input checked="" type="checkbox"/> Yes

APPENDIX A: Table Creation Script

The below scripts have been used to create the tables used in the examples in this document. The tables have already been created, so you do not need to run these scripts. This is for reference only.

Table Creation

```
REM *****
REM Create the REGIONS table to hold region information for locations REM
REM HR.LOCATIONS table has a foreign key to this table.

Prompt ***** Creating REGIONS table ....

CREATE TABLE regions
  ( region_id      NUMBER
    CONSTRAINT  region_id_nn NOT NULL
  , region_name    VARCHAR2(25)
  );

CREATE UNIQUE INDEX reg_id_pk
ON regions (region_id);

ALTER TABLE regions
ADD ( CONSTRAINT reg_id_pk
      PRIMARY KEY (region_id)
    ) ;

REM *****
REM Create the COUNTRIES table to hold country information for customers
REM and company locations.
REM OE.CUSTOMERS table and HR.LOCATIONS have a foreign key to this table.

Prompt ***** Creating COUNTRIES table ....
```

```
CREATE TABLE countries
( country_id      CHAR(2)
  CONSTRAINT country_id_nn NOT NULL
, country_name    VARCHAR2(40)
, region_id       NUMBER
, CONSTRAINT      country_c_id_pk
                  PRIMARY KEY (country_id)
)
ORGANIZATION INDEX;

ALTER TABLE countries
ADD ( CONSTRAINT countr_reg_fk
      FOREIGN KEY (region_id)
      REFERENCES regions(region_id)
) ;
```

```
REM *****
REM Create the LOCATIONS table to hold address information for company
departments.
REM HR.DEPARTMENTS has a foreign key to this table.
```

Prompt ***** Creating LOCATIONS table

```
CREATE TABLE locations
( location_id      NUMBER(4)
, street_address  VARCHAR2(40)
, postal_code     VARCHAR2(12)
, city            VARCHAR2(30)
  CONSTRAINT      loc_city_nn  NOT NULL
, state_province  VARCHAR2(25)
, country_id      CHAR(2)
) ;
```

```
CREATE UNIQUE INDEX loc_id_pk
ON locations (location_id) ;
```

```
ALTER TABLE locations
ADD ( CONSTRAINT loc_id_pk
      PRIMARY KEY (location_id)
, CONSTRAINT loc_c_id_fk
      FOREIGN KEY (country_id)
      REFERENCES countries(country_id)
) ;
```

```
Rem Useful for any subsequent addition of rows to locations table
Rem Starts with 3300
```

```
CREATE SEQUENCE locations_seq
START WITH      3300 INCREMENT BY 100
MAXVALUE        9900
NOCACHE
NOCYCLE;
```

```
REM *****
REM Create the DEPARTMENTS table to hold company department information.
REM HR.EMPLOYEES and HR.JOB_HISTORY have a foreign key to this table.
```

Prompt ***** Creating DEPARTMENTS table

```

CREATE TABLE departments
  ( department_id    NUMBER(4)
  , department_name  VARCHAR2(30)
    CONSTRAINT dept_name_nn  NOT NULL
  , manager_id      NUMBER(6)
  , location_id      NUMBER(4)
  ) ;

```

```

CREATE UNIQUE INDEX dept_id_pk
ON departments (department_id) ;

```

```

ALTER TABLE departments
ADD ( CONSTRAINT dept_id_pk
      PRIMARY KEY (department_id)
  , CONSTRAINT dept_loc_fk
      FOREIGN KEY (location_id)
        REFERENCES locations (location_id)
  ) ;

```

Rem Useful for any subsequent addition of rows to departments table Rem
Starts with 280

```

CREATE SEQUENCE departments_seq
  START WITH      280
  INCREMENT BY    10
  MAXVALUE        9990
  NOCACHE
  NOCYCLE;

```

```

REM *****
REM Create the JOBS table to hold the different names of job roles
REM within the company.
REM HR.EMPLOYEES has a foreign key to this table.

```

Prompt ***** Creating JOBS table

```

CREATE TABLE jobs
  ( job_id          VARCHAR2(10)
  , job_title       VARCHAR2(35)
    CONSTRAINT job_title_nn  NOT NULL
  , min_salary      NUMBER(6)
  , max_salary      NUMBER(6)
  ) ;

```

```

CREATE UNIQUE INDEX job_id_pk
ON jobs (job_id) ;

```

```

ALTER TABLE jobs
ADD ( CONSTRAINT job_id_pk
      PRIMARY KEY(job_id)
    ) ;
REM *****
REM Create the EMPLOYEES table to hold the employee personnel
REM information for the company. REM HR.EMPLOYEES has a self
REM referencing foreign key to this table.

```

```

Prompt ***** Creating EMPLOYEES table ....

```

```

CREATE TABLE employees
( employee_id      NUMBER(6)
, first_name       VARCHAR2(20)
, last_name        VARCHAR2(25)
  CONSTRAINT      emp_last_name_nn NOT NULL
, email            VARCHAR2(25)
  CONSTRAINT      emp_email_nn  NOT NULL
, phone_number     VARCHAR2(20)
, hire_date        DATE
  CONSTRAINT      emp_hire_date_nn NOT NULL
, job_id           VARCHAR2(10)
  CONSTRAINT      emp_job_nn   NOT NULL
, salary           NUMBER(8,2)
, commission_pct   NUMBER(2,2)
, manager_id       NUMBER(6) ,
department_id      NUMBER(4)
, CONSTRAINT      emp_salary_min
                  CHECK (salary > 0)
, CONSTRAINT      emp_email_uk
                  UNIQUE (email)
) ;

```

```

CREATE UNIQUE INDEX emp_emp_id_pk
ON employees (employee_id) ;

```

```

ALTER TABLE employees
ADD ( CONSTRAINT      emp_emp_id_pk
      PRIMARY KEY (employee_id)
    , CONSTRAINT      emp_dept_fk
      FOREIGN KEY (department_id)
      REFERENCES departments
    , CONSTRAINT      emp_job_fk
      FOREIGN KEY (job_id)
      REFERENCES jobs (job_id)

```

```

        , CONSTRAINT      emp_manager_fk
                          FOREIGN KEY (manager_id)
                          REFERENCES employees
    ) ;

```

```

ALTER TABLE departments
ADD ( CONSTRAINT dept_mgr_fk
      FOREIGN KEY (manager_id)
      REFERENCES employees (employee_id)
    ) ;

```

Rem Useful for any subsequent addition of rows to employees table
 Rem Starts with 207

```

CREATE SEQUENCE employees_seq
  START WITH      207
  INCREMENT BY    1
  NOCACHE
  NOCYCLE;

```

```

REM *****
REM Create the JOB_HISTORY table to hold the history of jobs that
REM employees have held in the past.
REM HR.JOBS, HR_DEPARTMENTS, and HR.EMPLOYEES have a foreign key to this
REM table.

```

Prompt ***** Creating JOB_HISTORY table

```

CREATE TABLE job_history
  ( employee_id  NUMBER(6)
    CONSTRAINT   jhist_employee_nn  NOT NULL
  , start_date   DATE
    CONSTRAINT   jhist_start_date_nn NOT NULL
  , end_date     DATE
    CONSTRAINT   jhist_end_date_nn  NOT NULL
  , job_id       VARCHAR2(10)
    CONSTRAINT   jhist_job_nn       NOT NULL
  , department_id NUMBER(4)
  , CONSTRAINT   jhist_date_interval
                CHECK (end_date > start_date)
  ) ;

```

```

CREATE UNIQUE INDEX jhist_emp_id_st_date_pk

```

```
ON job_history (employee_id, start_date) ;
```

```
ALTER TABLE job_history
ADD ( CONSTRAINT jhist_emp_id_st_date_pk
      PRIMARY KEY (employee_id, start_date)
    , CONSTRAINT jhist_job_fk
      FOREIGN KEY (job_id)
      REFERENCES jobs
    , CONSTRAINT jhist_emp_fk
      FOREIGN KEY (employee_id)
      REFERENCES employees
    , CONSTRAINT jhist_dept_fk
      FOREIGN KEY (department_id)
      REFERENCES departments
    ) ;
```

```
REM *****
REM Create the EMP_DETAILS_VIEW that joins the employees, jobs,  REM
departments, jobs, countries, and locations table to provide details REM
about employees.
```

```
Prompt ***** Creating EMP_DETAILS_VIEW view ...
```

```
CREATE OR REPLACE VIEW emp_details_view
(employee_id, job_id, manager_id, department_id,
location_id, country_id, first_name, last_name, salary,
commission_pct, department_name, job_title, city,
state_province, country_name, region_name)
AS SELECT
    e.employee_id,
    e.job_id,
    e.manager_id,
    e.department_id,
    d.location_id,
    l.country_id,
    e.first_name,
    e.last_name,
    e.salary,
    e.commission_pct,
    d.department_name,
    j.job_title,
    l.city,
    l.state_province,
    c.country_name,
```

```
    r.region_name
FROM
employees e,
departments d,
jobs j,
locations l,
countries c,
regions r
WHERE e.department_id = d.department_id
      AND d.location_id = l.location_id
      AND l.country_id = c.country_id
      AND c.region_id = r.region_id
      AND j.job_id = e.job_id
WITH READ ONLY;
COMMIT;
```

APPENDIX B: Table Creations Scripts for Bank Example

Below scripts have been used to create and populate data for the tables used in the Bank Example. You do not need to run the script, as the tables are already created. This is for reference only. The same is also available as, sql scripts named - SQL_BankTble.SQL

```

DROP TABLE TRANS_DTLS;

DROP TABLE TRANS_MSTR;

DROP TABLE CNTC_DTLS;

DROP TABLE ADDR_DTLS;

DROP TABLE ACCT_FD_CUST_DTLS;

DROP TABLE NOMINEE_MSTR;

DROP TABLE FD_DTLS;

DROP TABLE FD_MSTR;

DROP TABLE FDSLAB_MSTR;

DROP TABLE ACCT_MSTR; DROP
TABLE SPRT_DOC;

DROP TABLE CUST_MSTR;

DROP TABLE EMP_MSTR;

DROP TABLE BRANCH_MSTR;


-- BRANCH_MSTR

CREATE TABLE "BRANCH_MSTR"(
    "BRANCH_NO" VARCHAR2(10),
    "NAME" VARCHAR2(25));


-- EMP_MSTR

CREATE TABLE "EMP_MSTR"(
    "EMP_NO" VARCHAR2(10),
    "BRANCH_NO" VARCHAR2(10),
    "FNAME" VARCHAR2(25),
    "MNAME" VARCHAR2(25),
    "LNAME" VARCHAR2(25),
    "DEPT" VARCHAR2(30),
    "DESIG" VARCHAR2(30));


-- CUST_MSTR

CREATE TABLE "CUST_MSTR"(
    "CUST_NO" VARCHAR2(10),
    "FNAME" VARCHAR2(25),
    "MNAME" VARCHAR2(25),
    "LNAME" VARCHAR2(25),

```

```
"DOB_INC" DATE NOT NULL,
"OCCUP" VARCHAR2(25),
"PHOTOGRAPH" VARCHAR2(25),
"SIGNATURE" VARCHAR2(25),
"PANCOPY" VARCHAR2(1),
"FORM60" VARCHAR2(1));

-- SPRT_DOC
CREATE TABLE "SPRT_DOC" (
    "ACCT_CODE" VARCHAR2(4),
    "TYPE" VARCHAR2(40),
    "DOCS" VARCHAR2(75));

-- ACCT_MSTR
CREATE TABLE "ACCT_MSTR" (
    "ACCT_NO" VARCHAR2(10),
```

```

"SF_NO" VARCHAR2(10),
"LF_NO" VARCHAR2(10),
"BRANCH_NO" VARCHAR2(10),
"INTRO_CUST_NO" VARCHAR2(10),
"INTRO_ACCT_NO" VARCHAR2(10),
"INTRO_SIGN" VARCHAR2(1),
"TYPE" VARCHAR2(2),
"OPR_MODE" VARCHAR2(2),
"CUR_ACCT_TYPE" VARCHAR2(4),
"TITLE" VARCHAR2(30),
"CORP_CUST_NO" VARCHAR2(10),
"APLNDT" DATE,
"OPNDT" DATE,
"VERI_EMP_NO" VARCHAR2(10),
"VERI_SIGN" VARCHAR2(1),
"MANAGER_SIGN" VARCHAR2(1),
"CURBAL" NUMBER(8, 2) DEFAULT 0,
"STATUS" VARCHAR2(1) DEFAULT 'A');

```

-- FD_MSTR

```

CREATE TABLE "FD_MSTR"(
  "FD_SER_NO" VARCHAR2(10),
  "SF_NO" VARCHAR2(10),
  "BRANCH_NO" VARCHAR2(10),
  "INTRO_CUST_NO" VARCHAR2(10),
  "INTRO_ACCT_NO" VARCHAR2(10),
  "INTRO_SIGN" VARCHAR2(1),
  "ACCT_NO" VARCHAR2(10),
  "TITLE" VARCHAR2(30),
  "CORP_CUST_NO" VARCHAR2(10),
  "CORP_CNST_TYPE" VARCHAR(4),
  "VERI_EMP_NO" VARCHAR2(10),
  "VERI_SIGN" VARCHAR2(1),
  "MANAGER_SIGN" VARCHAR2(1));

```

-- FDSLAB_MSTR

```

CREATE TABLE "FDSLAB_MSTR"(
  "FDSLAB_NO" NUMBER(2),
  "MINPERIOD" NUMBER(5),
  "MAXPERIOD" NUMBER(5),
  "INTRATE" NUMBER(5,2));

```

-- FD_DTLS

```

CREATE TABLE "FD_DTLS"(
  "FD_SER_NO" VARCHAR2(10),
  "FD_NO" VARCHAR2(10),
  "TYPE" VARCHAR2(1),
  "PAYTO_ACCTNO" VARCHAR2(10),
  "PERIOD" NUMBER(5),
  "OPNDT" DATE,

```



```

        "DUEDT" DATE,
        "AMT" NUMBER(8,2),
        "DUEAMT" NUMBER(8,2),
        "INTRATE" NUMBER(3),
        "STATUS" VARCHAR2(1) DEFAULT 'A',
        "AUTO_RENEWAL" VARCHAR2(1));

-- ACCT_FD_CUST_DTLS
CREATE TABLE "ACCT_FD_CUST_DTLS"(
        "ACCT_FD_NO" VARCHAR2(10),          "CUST_NO" VARCHAR2(10));

-- NOMINEE_MSTR
CREATE TABLE "NOMINEE_MSTR"(
        "NOMINEE_NO" VARCHAR2(10),
        "ACCT_FD_NO" VARCHAR2(10),
        "NAME" VARCHAR2(75),
        "DOB" DATE,
        "RELATIONSHIP" VARCHAR2(25));

-- ADDR_DTLS
CREATE TABLE "ADDR_DTLS"(
        "ADDR_NO" NUMBER(6),
        "CODE_NO" VARCHAR2(10),
        "ADDR_TYPE" VARCHAR2(1),
        "ADDR1" VARCHAR2(50),
        "ADDR2" VARCHAR2(50),
        "CITY" VARCHAR2(25),
        "STATE" VARCHAR2(25),
        "PINCODE" VARCHAR2(6));

-- CNTC_DTLS
CREATE TABLE "CNTC_DTLS"(
        "ADDR_NO" NUMBER(6),
        "CODE_NO" VARCHAR2(10),
        "CNTC_TYPE" VARCHAR2(1),
        "CNTC_DATA" VARCHAR2(75));

-- TRANS_MSTR
CREATE TABLE "TRANS_MSTR"(
        "TRANS_NO" VARCHAR2(10),
        "ACCT_NO" VARCHAR2(10),
        "DT" DATE,
        "TYPE" VARCHAR2(1),
        "PARTICULAR" VARCHAR2(30),
        "DR_CR" VARCHAR2(1),
        "AMT" NUMBER(8,2),
        "BALANCE" NUMBER(8,2));

```

```

-- TRANS_DTLS
CREATE TABLE "TRANS_DTLS"(
    "TRANS_NO" VARCHAR2(10),
    "INST_NO" NUMBER(6),
    "INST_DT" DATE,
    "PAYTO" VARCHAR2(30),
    "INST_CLR_DT" DATE,
    "BANK_NAME" VARCHAR2(35),
    "BRANCH_NAME" VARCHAR2(25),
    "PAIDFROM" VARCHAR2(10));

-- Records for BRANCH_MSTR
INSERT INTO BRANCH_MSTR (BRANCH_NO, NAME) VALUES('B1', 'Vile Parle
(HO)');
INSERT INTO BRANCH_MSTR (BRANCH_NO, NAME) VALUES('B2', 'Andheri');
INSERT INTO BRANCH_MSTR (BRANCH_NO, NAME) VALUES('B3', 'Churchgate');
INSERT INTO BRANCH_MSTR (BRANCH_NO, NAME) VALUES('B4', 'Sion');
INSERT INTO BRANCH_MSTR (BRANCH_NO, NAME) VALUES('B5', 'Borivali');
INSERT INTO BRANCH_MSTR (BRANCH_NO, NAME) VALUES('B6', 'Matunga'); --
Records for EMP_MSTR
INSERT INTO EMP_MSTR (EMP_NO, BRANCH_NO, FNAME, MNAME, LNAME, DEPT,
DESIG)
    VALUES('E1', 'B1', 'Ivan', 'Nelson', 'Bayross', 'Administration',
'Managing Director');
INSERT INTO EMP_MSTR (EMP_NO, BRANCH_NO, FNAME, MNAME, LNAME, DEPT,
DESIG)
    VALUES('E2', 'B1', 'Amit', null, 'Desai', 'Loans And Financing',
'Head Of Dept. ');
INSERT INTO EMP_MSTR (EMP_NO, BRANCH_NO, FNAME, MNAME, LNAME, DEPT,
DESIG)
    VALUES('E3', 'B1', 'Maya', 'Mahima', 'Joshi', 'Accounts', 'Head Of
Dept. ');
INSERT INTO EMP_MSTR (EMP_NO, BRANCH_NO, FNAME, MNAME, LNAME, DEPT,
DESIG)
    VALUES('E4', 'B1', 'Peter', 'Iyer', 'Joseph', 'Client Servicing',
'Clerk');
INSERT INTO EMP_MSTR (EMP_NO, BRANCH_NO, FNAME, MNAME, LNAME, DEPT,
DESIG)
    VALUES('E5', 'B1', 'Mandhar', 'Dilip', 'Dalvi', 'Marketing', 'Head Of
Dept. ');
INSERT INTO EMP_MSTR (EMP_NO, BRANCH_NO, FNAME, MNAME, LNAME, DEPT,
DESIG)
    VALUES('E6', 'B1', 'Sonal', 'Abdul', 'Khan', 'Administration',
'Admin. Executive');
INSERT INTO EMP_MSTR (EMP_NO, BRANCH_NO, FNAME, MNAME, LNAME, DEPT,
DESIG)
    VALUES('E7', 'B1', 'Anil', 'Ashutosh', 'Kambli', 'Administration',
'Office Asst. ');
INSERT INTO EMP_MSTR (EMP_NO, BRANCH_NO, FNAME, MNAME, LNAME, DEPT,
DESIG)
    VALUES('E8', 'B1', 'Seema', 'P.', 'Apte', 'Client Servicing', 'Clerk');

```

```

INSERT INTO EMP_MSTR (EMP_NO, BRANCH_NO, FNAME, MNAME, LNAME, DEPT,
DESIG)
    VALUES('E9', 'B1', 'Vikram', 'Vilas', 'Randive', 'Accounts', 'Office
Asst.');
```

```

INSERT INTO EMP_MSTR (EMP_NO, BRANCH_NO, FNAME, MNAME, LNAME, DEPT,
DESIG)
    VALUES('E10', 'B1', 'Anjali', 'Sameer', 'Pathak', 'Marketing',
'Marketing Manager');
```

```

-- Records for CUST_MSTR
INSERT INTO CUST_MSTR (CUST_NO, FNAME, MNAME, LNAME, DOB_INC, OCCUP,
PHOTOGRAPH, SIGNATURE, PANCOPY, FORM60)
    VALUES('C1', 'Ivan', 'Nelson', 'Bayross', '25-JUN-1952', 'Self
Employed',
        'D:/ClntPht/C1.gif', 'D:/ClntSgnt/C1.gif', 'Y', 'Y');
```

```

INSERT INTO CUST_MSTR (CUST_NO, FNAME, MNAME, LNAME, DOB_INC, OCCUP,
PHOTOGRAPH, SIGNATURE, PANCOPY, FORM60)
    VALUES('C2', 'Chriselle', 'Ivan', 'Bayross', '29-OCT-1982',
'Service',
        'D:/ClntPht/C2.gif', 'D:/ClntSgnt/C2.gif', 'N', 'Y');
```

```

INSERT INTO CUST_MSTR (CUST_NO, FNAME, MNAME, LNAME, DOB_INC, OCCUP,
PHOTOGRAPH, SIGNATURE, PANCOPY, FORM60)
    VALUES('C3', 'Mamta', 'Arvind', 'Muzumdar', '28-AUG-1975',
'Service',
        'D:/ClntPht/C3.gif', 'D:/ClntSgnt/C3.gif', 'Y', 'Y');
```

```

INSERT INTO CUST_MSTR (CUST_NO, FNAME, MNAME, LNAME, DOB_INC, OCCUP,
PHOTOGRAPH, SIGNATURE, PANCOPY, FORM60)
    VALUES('C4', 'Chhaya', 'Sudhakar', 'Bankar', '06-OCT-1976',
'Service',
        'D:/ClntPht/C4.gif', 'D:/ClntSgnt/C4.gif', 'Y', 'Y');
```

```

INSERT INTO CUST_MSTR (CUST_NO, FNAME, MNAME, LNAME, DOB_INC, OCCUP,
PHOTOGRAPH, SIGNATURE, PANCOPY, FORM60)
    VALUES('C5', 'Ashwini', 'Dilip', 'Joshi', '20-NOV-1978', 'Business',
        'D:/ClntPht/C5.gif', 'D:/ClntSgnt/C5.gif', 'Y', 'Y');
```

```

INSERT INTO CUST_MSTR (CUST_NO, FNAME, MNAME, LNAME, DOB_INC, OCCUP,
PHOTOGRAPH, SIGNATURE, PANCOPY, FORM60)
    VALUES('C6', 'Hansel', 'I.', 'Colaco', '01-JAN-1982', 'Service',
        'D:/ClntPht/C6.gif', 'D:/ClntSgnt/C6.gif', 'N', 'Y');
```

```

INSERT INTO CUST_MSTR (CUST_NO, FNAME, MNAME, LNAME, DOB_INC, OCCUP,
PHOTOGRAPH, SIGNATURE, PANCOPY, FORM60)
    VALUES('C7', 'Anil', 'Arun', 'Dhone', '12-OCT-1983', 'Self
Employed',
        'D:/ClntPht/C7.gif', 'D:/ClntSgnt/C7.gif', 'N', 'Y');
```

```

INSERT INTO CUST_MSTR (CUST_NO, FNAME, MNAME, LNAME, DOB_INC, OCCUP,
PHOTOGRAPH, SIGNATURE, PANCOPY, FORM60)
    VALUES('C8', 'Alex', 'Austin', 'Fernandes', '30-SEP-1962',
'Executive',
        'D:/ClntPht/C8.gif', 'D:/ClntSgnt/C8.gif', 'Y', 'Y');
```

```

INSERT INTO CUST_MSTR (CUST_NO, FNAME, MNAME, LNAME, DOB_INC, OCCUP,
PHOTOGRAPH, SIGNATURE, PANCOPY, FORM60)
    VALUES('C9', 'Ashwini', 'Shankar', 'Apte', '19-APR-1979', 'Service',
        'D:/ClntPht/C9.gif', 'D:/ClntSgnt/C9.gif', 'Y', 'Y');
```

```

INSERT INTO CUST_MSTR (CUST_NO, FNAME, MNAME, LNAME, DOB_INC, OCCUP,
PHOTOGRAPH, SIGNATURE, PANCOPY, FORM60)
    VALUES('C10', 'Namita', 'S.', 'Kanade', '10-JUN-1978', 'Self
Employed',
        'D:/ClntPht/C10.gif', 'D:/ClntSgnt/C10.gif', 'Y', 'Y');
INSERT INTO CUST_MSTR (CUST_NO, FNAME, MNAME, LNAME, DOB_INC, OCCUP,
PHOTOGRAPH, SIGNATURE, PANCOPY, FORM60)
    VALUES('O11', null, null, null, '14-NOV-1997', 'Retail Business',
null, null, 'Y', 'N');
INSERT INTO CUST_MSTR (CUST_NO, FNAME, MNAME, LNAME, DOB_INC, OCCUP,
PHOTOGRAPH, SIGNATURE, PANCOPY, FORM60)
    VALUES('O12', null, null, null, '23-OCT-1992', 'Information
Technology', null, null, 'Y', 'N');
INSERT INTO CUST_MSTR (CUST_NO, FNAME, MNAME, LNAME, DOB_INC, OCCUP,
PHOTOGRAPH, SIGNATURE, PANCOPY, FORM60)
    VALUES('O13', null, null, null, '05-FEB-1989', 'Community
Welfare', null, null, 'Y', 'N');
INSERT INTO CUST_MSTR (CUST_NO, FNAME, MNAME, LNAME, DOB_INC, OCCUP,
PHOTOGRAPH, SIGNATURE, PANCOPY, FORM60)
    VALUES('O14', null, null, null, '24-MAY-1980', 'Retail Business',
null, null, 'N', 'Y');

-- Records for SPRT_DOC
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
    VALUES('0S', 'Individuals / Savings Bank Account', 'Driving Licence
/ Ration Card / Passport');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
    VALUES('0S', 'Individuals / Savings Bank Account', 'Birth
Certificate / School Leaving Certificate');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
    VALUES('1C', 'Propriety / Sole Trading Concerns', 'Letter From The
Propriety');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
    VALUES('2C', 'Partnership Concerns', 'Letter From The Partners');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)

```



```

VALUES('2C', 'Partnership Concerns',
'Partnership Deed /
Registration Certificate');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
VALUES('3C', 'Hindu Undivided Family Businesses', 'Letter From The
Karta');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
VALUES('3C', 'Hindu Undivided Family
Businesses', 'List Of
Members');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
VALUES('4C', 'Limited
Companies', 'Copy Of Board Of Directors''
Resolution For Opening The Account');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
VALUES('4C', 'Limited Companies',
'Memorandum and Articles Of
Association');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
VALUES('4C', 'Limited
Companies', 'Certificate Of Incorporation');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
VALUES('4C', 'Limited
Companies', 'Certificate Of Commencement Of
Business / Registration Certificate');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
VALUES('5C', 'Trust Accounts', 'Trust
Deed');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
VALUES('5C', 'Trust Accounts',
'Resolution Of Trustees');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
VALUES('5C', 'Trust Accounts', 'List Of
Trusties');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
VALUES('6C', 'Clubs / Societies',
'Resolution');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
VALUES('6C', 'Clubs /
Societies', 'Constitution And Bye-laws');

```

```

INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
                                VALUES('6C', 'Clubs /
Societies', 'Certificate Of Registration');
INSERT INTO SPRT_DOC (ACCT_CODE, TYPE, DOCS)
    VALUES('7C', 'Legislative Bodies', 'Letter From The Authority');
-- Records for ACCT_MSTR
INSERT INTO ACCT_MSTR (ACCT_NO, SF_NO, LF_NO, BRANCH_NO, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN, TYPE, OPR_MODE,
                                CUR_ACCT_TYPE, TITLE,
CORP_CUST_NO, APLNDT, OPNDT, VERI_EMP_NO,
VERI_SIGN, MANAGER_SIGN, CURBAL, STATUS)
    VALUES('SB1', 'SF-0001', 'JAN03-05', 'B1', 'C1', 'SB1', 'Y', 'SB',
'SI', '0S', null, null,
                                '05-JAN-2003', '05-JAN-2003',
'E1', 'Y', 'Y', 500, 'A');

INSERT INTO ACCT_MSTR (ACCT_NO, SF_NO, LF_NO, BRANCH_NO, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN, TYPE, OPR_MODE,
                                CUR_ACCT_TYPE, TITLE,
CORP_CUST_NO, APLNDT, OPNDT, VERI_EMP_NO, VERI_SIGN, MANAGER_SIGN,
CURBAL, STATUS)
                                VALUES('CA2', 'SF-0002',
'JAN03-10', 'B1', 'C1', 'SB1', 'Y', 'CA',
'JO', '1C', 'Uttam Stores', '011',
                                '07-JAN-2003', '10-JAN-2003',
'E1', 'Y', 'Y', 2000, 'A');

INSERT INTO ACCT_MSTR (ACCT_NO, SF_NO, LF_NO, BRANCH_NO, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN, TYPE, OPR_MODE,
                                CUR_ACCT_TYPE, TITLE,
CORP_CUST_NO, APLNDT, OPNDT, VERI_EMP_NO,
VERI_SIGN, MANAGER_SIGN, CURBAL, STATUS)
                                VALUES('SB3',
'SF0003', 'JAN03-22', 'B1', 'C4', 'SB3', 'Y', 'SB', 'SI', '0S', null,
null,
                                '20-JAN-2003', '22-JAN-2003',
'E4', 'Y', 'Y', 500, 'A');

INSERT INTO ACCT_MSTR (ACCT_NO, SF_NO, LF_NO, BRANCH_NO, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN, TYPE, OPR_MODE,
                                CUR_ACCT_TYPE, TITLE,
CORP_CUST_NO, APLNDT, OPNDT, VERI_EMP_NO,
VERI_SIGN, MANAGER_SIGN, CURBAL, STATUS)

```



```

VALUES('CA4', 'SF-0004',
'FEB03-05', 'B1', 'C4', 'SB3', 'Y', 'CA',
'AS', '4C', 'Sun''s Pvt. Ltd.', 'O12',
'02-FEB-2003', '05-FEB-2003',
'E4', 'Y', 'Y', 2000, 'A');
INSERT INTO ACCT_MSTR (ACCT_NO, SF_NO, LF_NO, BRANCH_NO, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN, TYPE, OPR_MODE,
CUR_ACCT_TYPE, TITLE,
CORP_CUST_NO, APLNDT, OPNDT, VERI_EMP_NO,
VERI_SIGN, MANAGER_SIGN, CURBAL, STATUS)
VALUES('SB5', 'SF-0005', 'FEB03-15', 'B1', 'C1', 'SB1', 'Y', 'SB',
'JO', 'OS', null, null,
'14-FEB-2003', '15-FEB-2003',
'E1', 'Y', 'Y', 500, 'A');
INSERT INTO ACCT_MSTR (ACCT_NO, SF_NO, LF_NO, BRANCH_NO, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN, TYPE, OPR_MODE,
CUR_ACCT_TYPE, TITLE,
CORP_CUST_NO, APLNDT, OPNDT, VERI_EMP_NO,
VERI_SIGN, MANAGER_SIGN, CURBAL, STATUS)
VALUES('SB6', 'SF-0006', 'FEB03-27', 'B1', 'C5', 'SB6', 'Y', 'SB',
'ES', 'OS', null, null,
'27-FEB-2003', '27-FEB-2003',
'E4', 'Y', 'Y', 500, 'A');
INSERT INTO ACCT_MSTR (ACCT_NO, SF_NO, LF_NO, BRANCH_NO, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN, TYPE, OPR_MODE,
CUR_ACCT_TYPE, TITLE,
CORP_CUST_NO, APLNDT, OPNDT, VERI_EMP_NO,
VERI_SIGN, MANAGER_SIGN, CURBAL, STATUS)
VALUES('CA7', 'SF-0007',
'MAR03-14', 'B1', 'C8', 'CA7', 'Y', 'CA',
'AS', '6C', 'Puru Hsg. Soc', 'O13',
'14-MAR-2003', '14-MAR-2003',
'E4', 'Y', 'Y', 2000, 'A');
INSERT INTO ACCT_MSTR (ACCT_NO, SF_NO, LF_NO, BRANCH_NO, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN, TYPE, OPR_MODE,
CUR_ACCT_TYPE, TITLE,
CORP_CUST_NO, APLNDT, OPNDT, VERI_EMP_NO,
VERI_SIGN, MANAGER_SIGN, CURBAL, STATUS)
VALUES('SB8', 'SF-0008', 'MAR03-29', 'B1', 'C9', 'SB8', 'Y', 'SB',
'SI', 'OS', null, null,
'27-MAR-2003', '29-MAR-2003',
'E1', 'Y', 'Y', 500, 'A');

```

```

INSERT INTO ACCT_MSTR (ACCT_NO, SF_NO, LF_NO, BRANCH_NO, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN, TYPE, OPR_MODE,
CUR_ACCT_TYPE, TITLE,
CORP_CUST_NO, APLNDT, OPNDT, VERI_EMP_NO,
VERI_SIGN, MANAGER_SIGN, CURBAL, STATUS)
VALUES('SB9', 'SF-0009', 'APR03-05', 'B1', 'C10', 'SB9', 'Y', 'SB',
'JO', 'OS', null, null,
'05-APR-2003', '05-APR-2003',
'E4', 'Y', 'Y', 500, 'A');

INSERT INTO ACCT_MSTR (ACCT_NO, SF_NO, LF_NO, BRANCH_NO, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN, TYPE, OPR_MODE,
CUR_ACCT_TYPE, TITLE,
CORP_CUST_NO, APLNDT, OPNDT, VERI_EMP_NO,
VERI_SIGN, MANAGER_SIGN, CURBAL, STATUS)
VALUES('CA10', 'SF-0010',
'APR03-19', 'B1', 'C10', 'SB9', 'Y', 'CA',
'AS', '3C', 'Ghar Karobar', 'O14',
'19-APR-2003', '19-APR-2003', 'E4', 'Y', 'Y', 2000, 'A');

-- Records for FD_MSTR

INSERT INTO FD_MSTR (FD_SER_NO, SF_NO, BRANCH_NO, ACCT_NO, TITLE,
CORP_CUST_NO, CORP_CNST_TYPE, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN, VERI_EMP_NO, VERI_SIGN, MANAGER_SIGN)
VALUES ('FS1', 'SF-0011', 'B1', 'CA2', 'Uttam Stores', 'O11', '1C',
null, null, 'N', 'E1', 'Y', 'Y');

INSERT INTO FD_MSTR (FD_SER_NO, SF_NO, BRANCH_NO, ACCT_NO, TITLE,
CORP_CUST_NO, CORP_CNST_TYPE, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN, VERI_EMP_NO, VERI_SIGN, MANAGER_SIGN)
VALUES ('FS2', 'SF-0012', 'B1', 'CA4', 'Sun''s Pvt. Ltd.', 'C12',
'4C', null, null, 'N', 'E1', 'Y', 'Y');

INSERT INTO FD_MSTR (FD_SER_NO, SF_NO, BRANCH_NO, ACCT_NO, TITLE,
CORP_CUST_NO, CORP_CNST_TYPE, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN,
VERI_EMP_NO, VERI_SIGN, MANAGER_SIGN)

```

```

VALUES ('FS3', 'SF-0013', 'B1', 'CA7', 'Puru Hsg. Soc', 'O13',
'6C', null, null, 'N', 'E4', 'Y', 'Y');
INSERT INTO FD_MSTR (FD_SER_NO, SF_NO, BRANCH_NO, ACCT_NO, TITLE,
CORP_CUST_NO, CORP_CNST_TYPE, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN, VERI_EMP_NO, VERI_SIGN, MANAGER_SIGN)
VALUES ('FS4', 'SF-0014', 'B1', 'CA10', 'Ghar Karobar', 'O14',
'3C', null, null, 'N', 'E4', 'Y', 'Y');
INSERT INTO FD_MSTR (FD_SER_NO, SF_NO, BRANCH_NO, ACCT_NO, TITLE,
CORP_CUST_NO, CORP_CNST_TYPE, INTRO_CUST_NO,
INTRO_ACCT_NO, INTRO_SIGN, VERI_EMP_NO, VERI_SIGN, MANAGER_SIGN)
VALUES ('FS5', 'SF-0015', 'B1', null, null, null, '0S', 'C7',
'SB6', 'Y', 'E4', 'Y', 'Y');

```

-- Record for FDSLAB_MSTR

```

INSERT INTO FDSLAB_MSTR (FDSLAB_NO, MINPERIOD, MAXPERIOD, INTRATE)
VALUES(1, 1, 30, 5);
INSERT INTO FDSLAB_MSTR (FDSLAB_NO, MINPERIOD, MAXPERIOD, INTRATE)
VALUES(2, 31, 92, 5.5);
INSERT INTO FDSLAB_MSTR (FDSLAB_NO, MINPERIOD, MAXPERIOD, INTRATE)
VALUES(3, 93, 183, 6);
INSERT INTO FDSLAB_MSTR (FDSLAB_NO, MINPERIOD, MAXPERIOD, INTRATE)
VALUES(4, 184, 365, 6.5);
INSERT INTO FDSLAB_MSTR (FDSLAB_NO, MINPERIOD, MAXPERIOD, INTRATE)
VALUES(5, 366, 731, 7.5);
INSERT INTO FDSLAB_MSTR (FDSLAB_NO, MINPERIOD, MAXPERIOD, INTRATE)
VALUES(6, 732, 1097, 8.5);
INSERT INTO FDSLAB_MSTR (FDSLAB_NO, MINPERIOD, MAXPERIOD, INTRATE)
VALUES(7, 1098, 1829, 10);

```

-- Record for FD_DTLS

```

INSERT INTO FD_DTLS (FD_SER_NO, FD_NO, TYPE, PAYTO_ACCTNO, PERIOD,
OPNDT,
DUEDT, AMT, DUEAMT,
INTRATE, STATUS, AUTO_RENEWAL)
VALUES('FS1', 'F1', 'S', 'CA2', 365, '02-APR-2003', '01-APR-2004',
5000, 5350.00, 6.5, 'A', 'Y');
INSERT INTO FD_DTLS (FD_SER_NO, FD_NO, TYPE, PAYTO_ACCTNO, PERIOD,
OPNDT,
DUEDT, AMT, DUEAMT,
INTRATE, STATUS, AUTO_RENEWAL)
VALUES('FS1', 'F2', 'S', 'CA2', 365, '02-APR-2003', '01-APR-2004',
5000, 5350.00, 6.5, 'A', 'N');
INSERT INTO FD_DTLS (FD_SER_NO, FD_NO, TYPE, PAYTO_ACCTNO, PERIOD,
OPNDT,
DUEDT, AMT, DUEAMT,
INTRATE, STATUS, AUTO_RENEWAL)
VALUES('FS2', 'F3', 'S', 'CA4', 366, '25-MAY-2003', '25-MAY-2004',
10000, 10802.19, 7.5, 'A', 'Y');
INSERT INTO FD_DTLS (FD_SER_NO, FD_NO, TYPE, PAYTO_ACCTNO, PERIOD,
OPNDT,
DUEDT, AMT, DUEAMT,

```

```

        INTRATE, STATUS, AUTO_RENEWAL)
        VALUES('FS2', 'F4', 'S', 'CA4', 366, '15-JUN-2003', '15-JUN-2004',
10000, 10802.19, 7.5, 'A', 'Y');
INSERT INTO FD_DTLS (FD_SER_NO, FD_NO, TYPE, PAYTO_ACCTNO, PERIOD, OPNDT,
DUEDT, AMT, DUEAMT,
        INTRATE, STATUS, AUTO_RENEWAL)
        VALUES('FS3', 'F5', 'S', 'CA7', 183, '24-JUN-2003', '24-DEC-2003',
2000, 2060.16, 6, 'A', 'Y');
INSERT INTO FD_DTLS (FD_SER_NO, FD_NO, TYPE, PAYTO_ACCTNO, PERIOD, OPNDT,
DUEDT, AMT, DUEAMT,
        INTRATE, STATUS, AUTO_RENEWAL)
        VALUES('FS4', 'F6', 'S', 'CA10', 732, '19-JUL-2003', '20-JUL-2005',
5000, 5902.47, 8.5, 'A', 'Y');

```

```

INSERT INTO FD_DTLS (FD_SER_NO, FD_NO, TYPE, PAYTO_ACCTNO, PERIOD, OPNDT,
DUEDT, AMT, DUEAMT,
                INTRATE, STATUS, AUTO_RENEWAL)
VALUES('FS5', 'F7', 'S', 'SB6', 366, '27-JUL-2003', '27-JUL-2004',
5000, 5401.10, 7.5, 'A', 'N');

-- Record for ACCT_FD_CUST_DTLS
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('SB1', 'C1');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('CA2', 'C2');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('CA2', 'C3');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('SB3', 'C4');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('CA4', 'C4');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('CA4', 'C5');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('SB5', 'C1');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('SB5', 'C4');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('SB6', 'C5');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('SB6', 'C7');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('CA7', 'C6');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('CA7', 'C8');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('SB8', 'C9');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('SB9', 'C3');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('SB9', 'C10');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('CA10',
'C10');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('CA10', 'C9');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('FS1', 'C2');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('FS1', 'C3');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('FS2', 'C4');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('FS2', 'C5');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('FS2', 'C5');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('FS3', 'C6');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('FS3', 'C8');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('FS4', 'C10');
INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('FS4', 'C9');

```

```

INSERT INTO ACCT_FD_CUST_DTLS (ACCT_FD_NO, CUST_NO) VALUES('FS5', 'C5');
-- Record for NOMINEE_MSTR

INSERT INTO NOMINEE_MSTR (NOMINEE_NO, ACCT_FD_NO, NAME, DOB,
RELATIONSHIP)
VALUES('N1', 'CA2', 'Joseph Martin Dias', '17-SEP-1984',
'Colleague');

INSERT INTO NOMINEE_MSTR (NOMINEE_NO, ACCT_FD_NO, NAME, DOB,
RELATIONSHIP)
VALUES('N2', 'CA2', 'Nilesh Sawant', '25-AUG-1987', 'Colleague');
INSERT INTO NOMINEE_MSTR (NOMINEE_NO, ACCT_FD_NO, NAME, DOB,
RELATIONSHIP)
VALUES('N3', 'SB1', 'Chriselle Ivan
Bayross', '25-JUN-1952',
'Daughter');

INSERT INTO NOMINEE_MSTR (NOMINEE_NO, ACCT_FD_NO, NAME, DOB,
RELATIONSHIP)
VALUES('N4', 'SB3', 'Mamta Arvind Muzumdar', '28-AUG-1975',
'Friend');

INSERT INTO NOMINEE_MSTR (NOMINEE_NO, ACCT_FD_NO, NAME, DOB,
RELATIONSHIP)
VALUES('N5', 'SB6', 'Preeti Suresh Shah', '12-FEB-1978', 'Friend');
INSERT INTO NOMINEE_MSTR (NOMINEE_NO, ACCT_FD_NO, NAME, DOB,
RELATIONSHIP)
VALUES('N6', 'SB8', 'Rohit Rajan
Sahakarkar', '30-MAY-1985',
'Relative');

```

```

INSERT INTO NOMINEE_MSTR (NOMINEE_NO, ACCT_FD_NO, NAME, DOB,
RELATIONSHIP)
    VALUES('N7', 'CA10', 'Namita S. Kanade', '10-JUN-1978', 'Niece');
INSERT INTO NOMINEE_MSTR (NOMINEE_NO, ACCT_FD_NO, NAME, DOB,
RELATIONSHIP)
    VALUES('N8', 'FS1', 'Rohit Rajan Sahakarkar', '30-MAY-1985',
'Relative');
INSERT INTO NOMINEE_MSTR (NOMINEE_NO, ACCT_FD_NO, NAME, DOB,
RELATIONSHIP)
    VALUES('N9', 'FS2', 'Joseph Martin Dias', '17-SEP-1984',
'Colleague');
INSERT INTO NOMINEE_MSTR (NOMINEE_NO, ACCT_FD_NO, NAME, DOB,
RELATIONSHIP)
    VALUES('N10', 'FS2', 'Nilesh Sawant', '25-AUG-1987', 'Colleague');
INSERT INTO NOMINEE_MSTR (NOMINEE_NO, ACCT_FD_NO, NAME, DOB,
RELATIONSHIP)
    VALUES('N11', 'FS3', 'Chriselle Ivan Bayross', '25-JUN-1952',
'Colleague');
INSERT INTO NOMINEE_MSTR (NOMINEE_NO, ACCT_FD_NO, NAME, DOB,
RELATIONSHIP)
    VALUES('N12', 'FS3', 'Mamta Arvind Muzumdar', '28-AUG-1975',
'Friend');
INSERT INTO NOMINEE_MSTR (NOMINEE_NO, ACCT_FD_NO, NAME, DOB,
RELATIONSHIP)
    VALUES('N13', 'FS4', 'Namita S. Kanade', '10-JUN-1978', 'Relative');
INSERT INTO NOMINEE_MSTR (NOMINEE_NO, ACCT_FD_NO, NAME, DOB,
RELATIONSHIP)
    VALUES('N14', 'FS5', 'Pramila P. Pius', '10-OCT-1985', 'Niece');
-- Record for ADDR_DTLS
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
    VALUES(1, 'B1', 'H', 'A/5, Jay Chambers,', 'Service Road, Vile Parle
(East)',,
        'Mumbai', 'Maharashtra', '400057');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
    VALUES(2, 'B2', 'B', 'BSES Chambers, 10th floor,',
        'Near Rly. Station, Andheri (West)',, 'Mumbai', 'Maharashtra',
'400058');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
    VALUES(3, 'B3', 'B', 'Prabhat Complex, No. 5 / 6,', 'Opp. Air India
Bldg., Churchgate,',
        'Mumbai', 'Maharashtra', '400004');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
    VALUES(4, 'B4', 'B', '23/A, Swarna Bldg., Smt. Rai Marg,',
        'Eastern Express Highway, Kurla (East)',, 'Mumbai',
'Maharashtra', '400045');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
    VALUES(5, 'B5', 'B', 'Vikas Centre, Shop 37, Near National Park,',

```

```
        'Western Express Highway, Borivali (East)', 'Mumbai',  
'Maharashtra', '400078');  
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,  
STATE, PINCODE)  
VALUES(6, 'B6', 'B', '24/A, Mahim Plaza, First Floor,', 'Mahim (West)',  
        'Mumbai', 'Maharashtra', '400016');
```



```

INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
VALUES(7, 'E1', 'N', 'F-12, Diamond
Palace, West Avenue,',
'North Avenue, Santacruz (West)', 'Mumbai', 'Maharashtra', '400056');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
VALUES(8, 'E2', 'C', 'Desai House,
Plot No. 25, P.G. Marg,',
'Near Malad Rly. Stat., Malad
(West)', 'Mumbai',
'Maharashtra', '400078');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
VALUES(9, 'E3', 'N', 'Room No. 56,
3rd Floor, Swamibhavan,',
'J. P. Road Junction, Andheri
(East)', 'Mumbai',
'Maharashtra', '400059');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
VALUES(10, 'E4', 'C', '301, Thomas Palace, Opp. Indu Child Care,',
'Yadnik Nagar, Andheri (West)', 'Mumbai', 'Maharashtra', '400058');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
VALUES(11, 'E5', 'C', '456/A, Bldg.
No. 4, Vahatuk Nagar,',
'Amboli, Andheri (West)', 'Mumbai', 'Maharashtra', '400058');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
VALUES(12, 'E6', 'N', '201,
Meena Towers, Nr. Sun Gas Agency,',
'S. V. Rd., Goregoan (West)', 'Mumbai', 'Maharashtra', '400076');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
VALUES(13, 'E7', 'N', 'Patel
Chawl, Rm. No. 15, B. P. Lal Marg,',
'Mahim (West)', 'Mumbai',
'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
VALUES(14, 'E8', 'C', 'A - 10, Neelam, L. J. Road,', 'Mahim
(East)',
'Mumbai', 'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,

```

```

STATE, PINCODE)
VALUES(15, 'E9', 'N', '1/12 Bal
Govindas Society, M. B. Raut Rd.,',
'Dadar (East)', 'Mumbai',
'Maharashtra', '400028');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
VALUES(16, 'E10', 'C', 'Pathak Nagar, Cadal Road,', 'Mahim (West)', 'Mumbai',
'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
VALUES(17, 'C1', 'C', 'F-12, Diamond
Palace, West Avenue,',
'North Avenue, Santacruz (West)', 'Mumbai', 'Maharashtra', '400056');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
VALUES(18, 'C2', 'C', 'F-12, Silver Stream,', 'Santacruz (East)', 'Mumbai',
'Maharashtra', '400056');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)

```

```

VALUES(19, 'C3', 'C', 'Magesh Prasad,', 'Saraswati Baug,
Jogeshwari(E)',,
        'Mumbai', 'Maharashtra', '400060');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
        VALUES(20, 'C4', 'C', '4, Sampada,', 'Kataria Road, Mahim,',
'Mumbai',
        'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
        VALUES(21, 'C5', 'C', '104, Vikram Apts. Bhagat Lane,', 'Shivaji
Park, Mahim,',
        'Mumbai', 'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
        VALUES(22, 'C6', 'C', '12, Radha Kunj, N.C Kelkar Road,', 'Dadar,',
'Mumbai',
        'Maharashtra', '400028');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
        VALUES(23, 'C7', 'C', 'A/14, Shanti Society, Mogal Lane,', 'Mahim,',
'Mumbai',
        'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
        VALUES(24, 'C8', 'C', '5, Vagdevi, Senapati Bapat Rd.', 'Dadar,',
'Mumbai',
        'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
        VALUES(25, 'C9', 'C', 'A-10 Nutan Vaishali,', 'Shivaji Park,
Mahim,', 'Mumbai',
        'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
        VALUES(26, 'C10', 'C', 'B-10, Makarand Society,', 'Cadal Road,
Mahim,', 'Mumbai',
        'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
        VALUES(27, 'N1', 'C', '307/E, Meena Mansion,', 'R. S. Road, Andheri
(West)',,
        'Mumbai', 'Maharashtra', '400058');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
        VALUES(28, 'N2', 'C', 'Smt. Veenu Chawl, Sawant Colony Rd.',,
        'Opp. Veer Road, Matunga (West)',, 'Mumbai', 'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
        VALUES(29, 'N3', 'C', 'F-12, Silver Stream,', 'Santacruz (East)',,
'Mumbai',
        'Maharashtra', '400056');

```

```
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,  
STATE, PINCODE)  
VALUES(30, 'N4', 'C', 'Magesh Prasad,', 'Saraswati Baug,  
Jogeshwari(E)', '  
Mumbai', 'Maharashtra', '400060');  
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,  
STATE, PINCODE)
```

```

VALUES(31, 'N5', 'C', 'Rita Apartment, Room No. 46, 2nd Floor,',
      'J. P. Road, Andheri (East)', 'Mumbai', 'Maharashtra',
      '400067');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
      VALUES(32, 'N6', 'N', '106/A, Sunrise Apmt., Opp. Vahatuk Nagar,',
      'Kevni-Pada, Jogeshwari (West)', 'Mumbai', 'Maharashtra', '400102');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
      VALUES(33, 'N7', 'C', 'Pathak Nagar, Cadal Road,', 'Mahim (West)',
      'Mumbai',
      'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
      VALUES(34, 'O11', 'H', 'Shop No. 4, Simon Streams,',
      'V. P. Road, Andheri (West)', 'Mumbai', 'Maharashtra',
      '400058');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
      VALUES(35, 'O12', 'H', '230-E, Patel Chambers,', 'Service Road, Vile
Parle (East)',
      'Mumbai', 'Maharashtra', '400057');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
      VALUES(36, 'O13', 'H', 'G-2, Puru Hsg. Society,', 'Senapati Bapat Rd.,
Dadar',
      'Mumbai', 'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
      VALUES(37, 'O14', 'H', 'B-10, Makarand Society,', 'Cadale Road, Mahim',
      'Mumbai', 'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
      VALUES(38, 'N8', 'N', '106/A, Sunrise Apmt., Opp. Vahatuk Nagar,',
      'Kevni-Pada, Jogeshwari (West)', 'Mumbai', 'Maharashtra', '400102');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
      VALUES(39, 'N9', 'C', '307/E, Meena Mansion,', 'R. S. Road, Andheri
(West)',
      'Mumbai', 'Maharashtra', '400058');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
      VALUES(40, 'N10', 'C', 'Smt. Veenu Chawl, Sawant Colony Rd.,',
      'Opp. Veer Road, Matunga (West)', 'Mumbai', 'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
      VALUES(41, 'N11', 'C', 'F-12, Silver Stream,', 'Santacruz (East)',
      'Mumbai',
      'Maharashtra', '400056');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
      VALUES(42, 'N12', 'C', 'Magesh Prasad', 'Saraswati Baug,

```

```

Jogeshwari(E)',',
        'Mumbai', 'Maharashtra', '400060');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
        VALUES(43, 'N13', 'C', 'Pathak Nagar, Cadal Road,', 'Mahim (West)',',
'Mumbai',
        'Maharashtra', '400016');
INSERT INTO ADDR_DTLS (ADDR_NO, CODE_NO, ADDR_TYPE, ADDR1, ADDR2, CITY,
STATE, PINCODE)
        VALUES(44, 'N14', 'C', '405,Vahatuk Nagar, Kevni-Pada,', 'Jogeshwari
(West)',', 'Mumbai',
        'Maharashtra', '400102');

-- Record for CNTC_DTLS
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(1,
'B1', 'O', '26124571');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(1,
'B1', 'F', '26124533');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(1,
'B1', 'E', 'admin_vileparle@bom2.vsnl.in');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(2,
'B2', 'O', '26790014');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(2,
'B2', 'E', 'admin_andheri@bom2.vsnl.in');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(3,
'B3', 'O', '23457855');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(3,
'B3', 'E', 'admin_churchgate@bom2.vsnl.in');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(4,
'B4', 'O', '25545455');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(4,
'B4', 'E', 'admin_sion@bom2.vsnl.in');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(5,
'B5', 'O', '28175454');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(5,
'B5', 'E', 'admin_borivali@bom2.vsnl.in');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(6,
'B6', 'O', '24304545');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(6,
'B6', 'E', 'admin_matunga@bom2.vsnl.in');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(7,
'E1', 'R', '26045953');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(8,
'E2', 'R', '28883779');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES(9,
'E3', 'R', '28377634');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(10, 'E4', 'R', '26323560');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(11, 'E5', 'R', '26793231');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(12, 'E6', 'R', '28085654');

```

```

INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(13, 'E7', 'R', '24442342');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(14, 'E8', 'R', '24365672');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(15, 'E9', 'R', '24327349');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(16, 'E10', 'R', '24302579');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(17, 'C1', 'R', '26405853');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(17, 'C1', 'O', '26134553');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(17, 'C1', 'O', '26134571');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(17, 'C1', 'M', '9820178955');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(18, 'C2', 'R', '26045754');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(18, 'C2', 'O', '26134571');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(19, 'C3', 'R', '28324567');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(19, 'C3', 'O', '26197654');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(20, 'C4', 'R', '24449852');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(20, 'C4', 'O', '28741370');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(21, 'C5', 'R', '24302934');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(21, 'C5', 'O', '22819964');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(22, 'C6', 'R', '24217592');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(23,
'C7', 'R', '24372247');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(24, 'C8', 'O', '26480903');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(25, 'C9', 'R', '24313408');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(25,
'C9', 'M', '9821176651');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(26, 'C10', 'R', '24362680');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(26, 'C10', 'O', '28973355');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(26,
'C10', 'M', '9820484648');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA)
VALUES(27, 'N1', 'R', '26762154');

```

```

INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (28,
'N2', 'R', '24307887');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (29,
'N3', 'R', '260455754');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (30,
'N4', 'R', '28645489');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (31,
'N5', 'R', '30903564');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (32,
'N6', 'R', '26793771');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (33,
'N7', 'R', '24304455');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (34,
'O11', 'O', '26790055');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (34,
'O11', 'F', '26784409');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (35,
'O12', 'O', '26120455');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (35,
'O12', 'O', '26120456');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (35,
'O12', 'F', '26121450');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (35,
'O12', 'E', 'admin@sunpvtltd.com');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (35,
'O12', 'W', 'www.sunpvtltd.com');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (36,
'O13', 'O', '24301090');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (36,
'O13', 'O', '24301196');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (37,
'O14', 'O', '24321122');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (38,
'N8', 'R', '26793771');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (39,
'N9', 'R', '26762154');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (40,
'N10', 'R', '24307887');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (41,
'N11', 'R', '260455754');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (42,
'N12', 'R', '28645489');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (43,
'N13', 'R', '24304455');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (44,
'N14', 'R', '26790180');
INSERT INTO CNTC_DTLS (ADDR_NO, CODE_NO, CNTC_TYPE, CNTC_DATA) VALUES (44,
'N14', 'R', '26771275');

-- Record for TRANS_MSTR
INSERT INTO TRANS_MSTR (TRANS_NO, ACCT_NO, DT, TYPE, PARTICULAR, DR_CR,
AMT, BALANCE)

```



```

VALUES('T1', 'SB1', '05-JAN-2003', 'C', 'Initial Payment', 'D', 500,
500);
INSERT INTO TRANS_MSTR (TRANS_NO, ACCT_NO, DT, TYPE, PARTICULAR, DR_CR,
AMT, BALANCE)
VALUES('T2', 'CA2', '10-JAN-2003', 'C', 'Initial Payment', 'D',
2000, 2000);
INSERT INTO TRANS_MSTR (TRANS_NO, ACCT_NO, DT, TYPE, PARTICULAR, DR_CR,
AMT, BALANCE)
VALUES('T3', 'SB3', '22-JAN-2003', 'C', 'Initial Payment', 'D', 500,
500);
INSERT INTO TRANS_MSTR (TRANS_NO, ACCT_NO, DT, TYPE, PARTICULAR, DR_CR,
AMT, BALANCE)
VALUES('T4', 'CA4', '05-FEB-2003', 'B', 'Initial Payment', 'D',
2000, 2000);
INSERT INTO TRANS_MSTR (TRANS_NO, ACCT_NO, DT, TYPE, PARTICULAR, DR_CR,
AMT, BALANCE)
VALUES('T5', 'SB5', '15-FEB-2003', 'B', 'Initial Payment', 'D', 500,
500);
INSERT INTO TRANS_MSTR (TRANS_NO, ACCT_NO, DT, TYPE, PARTICULAR, DR_CR,
AMT, BALANCE)
VALUES('T6', 'SB6', '27-FEB-2003', 'C', 'Initial Payment', 'D', 500,
500);
INSERT INTO TRANS_MSTR (TRANS_NO, ACCT_NO, DT, TYPE, PARTICULAR, DR_CR,
AMT, BALANCE)
VALUES('T7', 'CA7', '14-MAR-2003', 'B', 'Initial Payment', 'D',
2000, 2000);
INSERT INTO TRANS_MSTR (TRANS_NO, ACCT_NO, DT, TYPE, PARTICULAR, DR_CR,
AMT, BALANCE)
VALUES('T8', 'SB8', '29-MAR-2003', 'C', 'Initial Payment', 'D', 500,
500);

```

```

INSERT INTO TRANS_MSTR (TRANS_NO, ACCT_NO, DT, TYPE, PARTICULAR, DR_CR,
AMT, BALANCE)
    VALUES('T9', 'SB9', '05-APR-2003', 'C', 'Initial Payment', 'D', 500,
500);
INSERT INTO TRANS_MSTR (TRANS_NO, ACCT_NO, DT, TYPE, PARTICULAR, DR_CR,
AMT, BALANCE)
    VALUES('T10', 'SB9', '15-APR-2003', 'B', 'CLR-204907', 'D', 3000,
3500);
INSERT INTO TRANS_MSTR (TRANS_NO, ACCT_NO, DT, TYPE, PARTICULAR, DR_CR,
AMT, BALANCE)
    VALUES('T11', 'SB9', '17-APR-2003', 'C', 'Self', 'W', 2500, 1000);
INSERT INTO TRANS_MSTR (TRANS_NO, ACCT_NO, DT, TYPE, PARTICULAR, DR_CR,
AMT, BALANCE)
    VALUES('T12', 'CA10', '19-APR-2003', 'B', 'Initial Payment', 'D',
2000, 2000);
INSERT INTO TRANS_MSTR (TRANS_NO, ACCT_NO, DT, TYPE, PARTICULAR, DR_CR,
AMT, BALANCE)
    VALUES('T13', 'SB9', '05-JUN-2003', 'B', 'CLR-204908', 'D', 3000,
4000);
INSERT INTO TRANS_MSTR (TRANS_NO, ACCT_NO, DT, TYPE, PARTICULAR, DR_CR,
AMT, BALANCE)
    VALUES('T14', 'SB9', '27-JUN-2003', 'C', 'Self', 'W', 2500, 1500);
-- Record for TRANS_DTLS
INSERT INTO TRANS_DTLS (TRANS_NO, INST_NO, INST_DT, PAYTO, INST_CLR_DT,
BANK_NAME, BRANCH_NAME, PAIDFROM)
    VALUES('T4', 098324, '02-FEB-2003', 'Self', '05-FEB-2003', 'HDFC',
'Vile Parle (East)', '2982');
INSERT INTO TRANS_DTLS (TRANS_NO, INST_NO, INST_DT, PAYTO, INST_CLR_DT,
BANK_NAME, BRANCH_NAME, PAIDFROM)
    VALUES('T5', 232324, '14-FEB-2003', 'Self', '15-FEB-2003', 'India
Bank', 'Andheri (West)', '30434');
INSERT INTO TRANS_DTLS (TRANS_NO, INST_NO, INST_DT, PAYTO, INST_CLR_DT,
BANK_NAME, BRANCH_NAME, PAIDFROM)
    VALUES('T7', 434560, '14-MAR-2003', 'Self', '14-MAR-2003', 'ICICI
Bank', 'Bandra (West)', '4882');
INSERT INTO TRANS_DTLS (TRANS_NO, INST_NO, INST_DT, PAYTO, INST_CLR_DT,
BANK_NAME, BRANCH_NAME, PAIDFROM)
    VALUES('T10', 204907, '14-APR-2003', 'Self', '15-APR-2003', 'Memon
Co-operative Bank', 'Jogeshwari (West)', '1767');
INSERT INTO TRANS_DTLS (TRANS_NO, INST_NO, INST_DT, PAYTO, INST_CLR_DT,
BANK_NAME, BRANCH_NAME, PAIDFROM)
    VALUES('T12', 100907, '19-APR-2003', 'Self', '19-APR-2003', 'Memon
Co-operative Bank', 'Jogeshwari (West)', '2001');
INSERT INTO TRANS_DTLS (TRANS_NO, INST_NO, INST_DT, PAYTO, INST_CLR_DT,

```

```
BANK_NAME, BRANCH_NAME, PAIDFROM)
VALUES('T13', 204908, '01-JUN-2003', 'Self', '05-JUN-2003', 'Memon
Co-operative Bank', 'Jogeshwari (West)', '1767');

COMMIT;
```

APPENDIX C: Further Reading

1. Oracle® Developer Suite Release Notes 10g Release 2 (10.1.2) for Solaris Operating System (SPARC), Microsoft Windows, and Linux x86 , Part No - B16010-05
2. Oracle® Reports Developer
Building Reports
Release 6i
January, 2000
Part No. A73172-01
3. Oracle Reports 10g: Build Reports: Oracle Curriculum—Oracle Developer Suite (ODS) Forms & Reports Section, Revision 11
by Sideris Courseware Corporation
Available on TCS Books 24x7 (Ultimatix > Learning & Collaboration > Book 24x7
URL: <http://www.books24x7.com/toc.aspx?bookid=42371> (available through ultimatix only)
4. Oracle Reports 10g: Build Reports
By: Sideris Courseware Corp.
Available on TCS Books 24x7 (Ultimatix > Learning & Collaboration > Safari Books Online)