ORACLE REPORT BUILDER

What is a Report ?

Report is a mean/medium through which we extract existing data and display them as per our business requirement.

Why do we need a Report?

a. To Check the Correctness of Data

b. For Business Analysis

c. For Statutory Requirement

Different means to generate a report in Oracle

- Report Developer (Reports 6i)
- SQL*Reports
- Writing into a file through PL/SQL

What is Reports Developer?

Reports Developer is a powerful enterprise reporting tool used to build reports that dynamically retrieve, format, and distribute information stored in the database.

Advantages of Report Developer

- Web publishes high quality reports.
- Perform unlimited data formatting.
- Take advantage of application server based reporting.
- User Friendly

Understanding the Parts of a Report

A report is made up of objects, the building blocks that define the various parts of the report. Some objects, like fields, are visible in the report and others, like queries, work behind the scenes.

The look and feel of a report is determined by two models:

- Data model.
- Layout model.

Data Model

The data model is composed of objects that define the data to be included in a report. Data model objects include:

- Queries
- Columns
- Groups
- Parameters
- Data links

Queries:

Queries select the data from a standard data source such as Oracle, DB2, or SQL/DS, using SQL SELECT or PL/SQL statements.

Columns:

Columns represent the columns in the datasource that are selected by the query. You can also define columns based on computed values such as summaries (also called totals) or values set conditionally.

Groups:

Groups organize the data into sets and hierarchies. By default, one group is created for each query in the report. The group contains all the columns selected by the query.

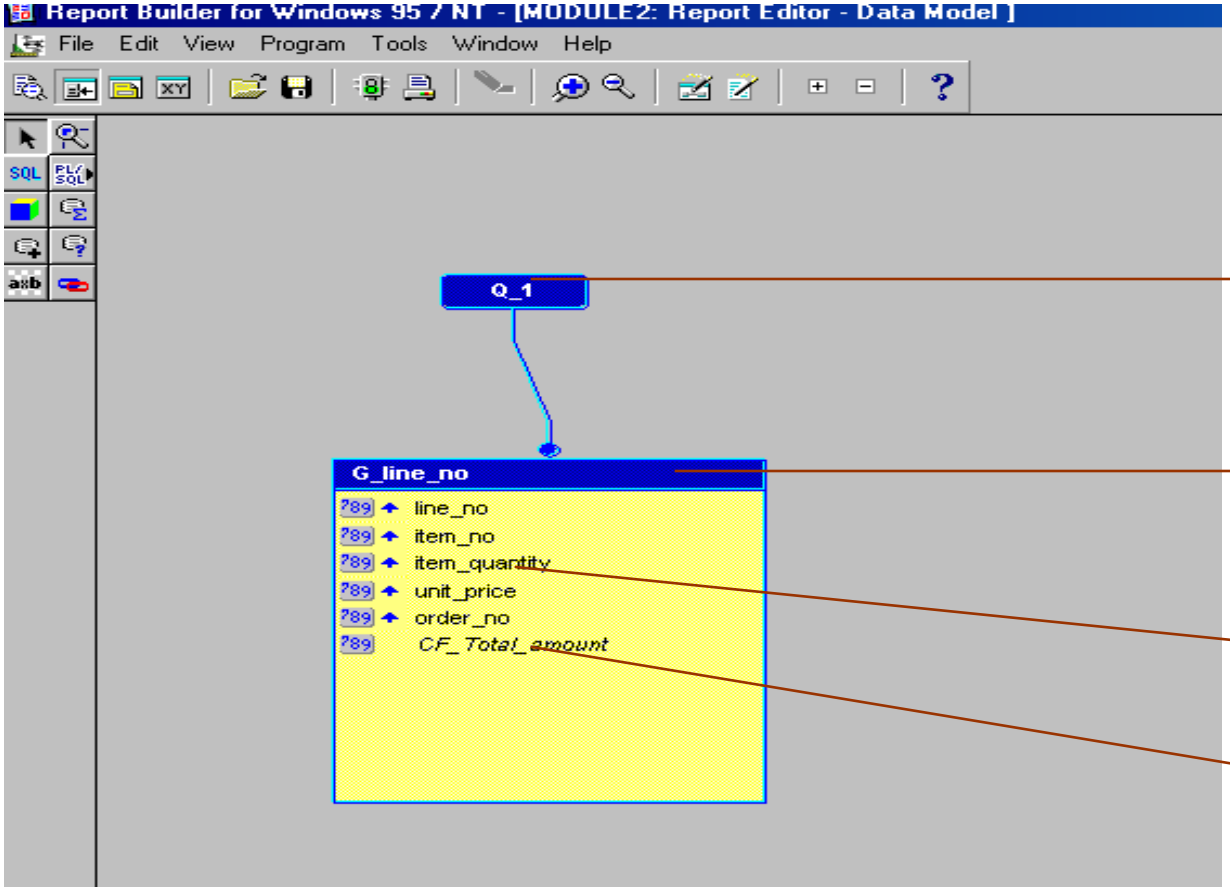
Parameters:

Parameters are variables, such as the printer name, to which users can assign values when the report is run.

Data links:

Data links are used to define a master/detail (parent-child) relationship between the data of two queries.

Structure of the Data Model



Layout Models:

The layout model is composed of objects that define the positioning and appearance of data and other objects in a report. This objects include:

- Repeating Frames
- Frames
- Fields
- Anchors
- Boilerplate Objects
- Buttons

Repeating Frames

Repeating frames are used to display the rows of data that are retrieved for a group. Thus, they "repeat" until all the data is retrieved.

Frames

Frames are used to keep layout objects together and can be used to protect layout objects from being overwritten by repeating frames at runtime.

Fields

Fields define how columns appear in a report, such as the format of currency amounts and dates.

Anchors:

Anchors are used to determine the relative positioning of one object to another in a report. The anchor attaches the anchored object, or child, to an anchoring object, or parent.

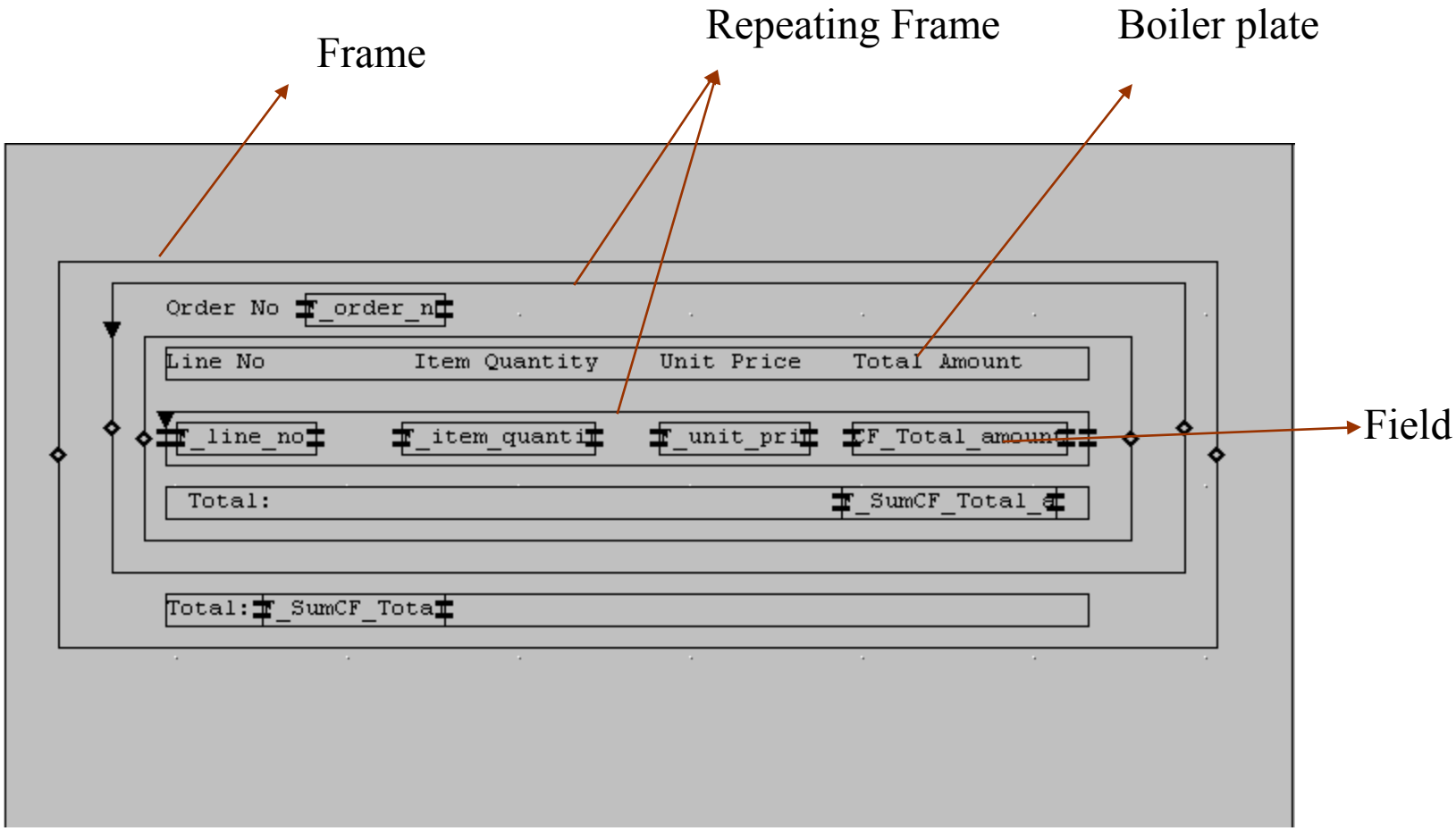
Boilerplate Objects:

Boilerplate objects are objects that appear in the report each time the report is run. Examples of boilerplate objects are field labels, graphics etc.

Buttons:

Buttons are objects users can click in an online report to display videos, sounds, or images, or execute a PL/SQL program that performs some action, such as launching another report.

Structure of the Layout Model



Formula Column

- Performs a user-defined computation
- Executes a PL/SQL function
- Must return a value
- Can be a Character, Number, Date
- Returned value must match datatype

```
function CF_SALCALCFormula return Number is
begin
  return(my_function :salary)
end;
```

Summary Column

- Specific properties:
  - Function
  - Source
  - Reset At
  - Compute At
- Datatype depends on Source datatype.

Data Model: Parameters

- To create and use the user parameters.
- Lexical vs Bind references.
- List of Values
- To use the system parameters.
- Build a parameter form

Data Model: Parameters: User Parameter

Bind reference replaces a value

- :parameter_name

parameter object is created by default

Lexical reference replaces a clause

- &parameter_name

parameter object is never created by default

List of Values

- Steps to create static LOVs
  - In the parameter property palette, choose the LOV property. The static values radio button is selected by default.
  - Enter a value in the value field and choose ADD
  - Repeat for each value you want in the list
  - Note: To remove a value, select the value in the list and choose REMOVE.
- Steps to create dynamic LOVs
  - In the parameter property palette, choose the LOV property.
  - Choose SELECT statement. The SQL query statement area displays.
  - Enter query to populate the list of values. You can include more than one column; the parameter takes its value from the first column in the list.
  - Set the restrict list to predetermined values property, as required.

Global Report Triggers

Report Builder has five global report triggers.

- Before Report
- After Report
- Between Pages
- Before Parameter
- After Parameter

Before Report Trigger

Before Report Fires before the report is executed but after queries are parsed.

After Report Trigger

After Report Fires after you exit the Previewer, or after report output is sent to a specified destination.

Between Pages Trigger

Between Pages Fires before each page of the report is formatted, except the very first page.

Before Parameter Form

Before Parameter Form Fires before the Runtime Parameter Form is displayed.

After Parameter Form

After Parameter Form Fires after the Runtime Parameter Form is displayed.

Format Trigger

This trigger is used to conditionally format and print data on the report.

Steps to be followed to design a report using report wizard :

- Select a report style and enter the title
- Build the query
- Group any field if required
- Select the fields to be displayed in the report
- Select totals to be displayed, if required
- Change the labels as required
- Select a template
- Click the finish button
- Save the module

Steps to be followed to design a report using manual method :

- Build the query in data model.
- Break the group if necessary
- Create formula columns, summary columns, if necessary
- Switch to layout model
- Create frames, repeating frames, fields, boiler plate objects as required.
- Compile and run the report
- Save the report module

SRW Package: Outputting Message

`SRW.MESSAGE` : This procedure displays a message with the message number and text that you specify. The message is displayed in the format below. After the message is raised and you accept it, the report execution will continue.

Exception

```
WHEN <exception> THEN
 SRW.MESSAGE(999, 'Warning: report continues');
```

Compilation of PL/SQL code within the Report :

Program → Compile → All or Incremental

Save Report Module :

1. File --> Save
2. Save Icon
3. Enter Filename
4. Save to the File System

Storage type of Report builder Modules

RDF – Report builder module

REP – Report executable module

REX – Report stored in ASCII file

Report output can be saved as :

- PDF
- RTF
- HTML
- XML
- Postscript
- Delimited

Report 1:

List out all customers with the following information:

- Order No
- Item Name
- Ship to location

Sample Output

Order no	Item Name	Ship To Location
1001	Floppy	101, Street 1, Kolkata - 700091
1002	CD Rider	102, Street 2, Kolkata - 700091
1003	CPU	103, Street 3, Kolkata - 700091
1004	Pen Drive	104, Street 4, Kolkata - 700091
1005	Mouse	105, Street 5, Kolkata - 700091

Report 2:

List out all orders with the following details:

1. Order #
2. Line #
3. Item #
4. Unit Price
5. Item Quantity
6. Total Price (Unit Price * Item Quantity)
7. Grand Total

Sample Output

Order #	Item Name	Item #	Unit Price	Quantity	Total Price
11	a	101	10.00	2	20.00
11	b	102	15.00	5	75.00
12	c	103	20.00	10	200.00
12	d	104	30.00	5	150.00
12	e	105	40.00	10	400.00
				Grand Total	**845.00**

Report 3: List out Orders by Grouping Orders Shown in Report 2.

		Order Details		Page: 1
				Date: 12/03/2004
Order #	**11**			
Line #	**Item #**	**Unit Price**	**Quantity**	**Total Price**
1	101	10.00	2	20.00
2	102	15.00	5	75.00
Order Total				**95.00**
Order #	**12**			
1	103	20.00	10	200.00
2	104	30.00	5	150.00
3	105	40.00	10	400.00
Order Total				**750.00**

Report 4: List out Order details for a specific order.

		Order Details		Page: 1
				Date: 12/03/2004
Order #	11			
Line #	Item #	Unit Price	Quantity	Total Price
1	101	10.00	2	20.00
2	102	15.00	5	75.00
Order Total				95.00

END