

ORACLE
SQL , PLSQL ,
APEX
How To's

BY

DR.S.RAGHUNATHAN

MCOM, DLLAL, PGDPM, CCCP, DSADP,

MPHIL, AICWA, PHD

ABOUT THE AUTHOR



Dr.S.RAGHUNATHAN is one of the best trainers in various software's and trained newbie's and professionals. He is an expert in clipper and developed various application software's for his thesis purpose and these applications software's aged 20 years and still alive on usage. He is professionally qualified as Cost Accountant. Now that, he starts writing about his experience on teaching to cater the needs of his second innings (after retirement) of his life.

ACKNOWLEDGMENTS

This book is dedicated to my DAD



(A V Sowrirajan (92 years) Retired Hindi cum
Sanskrit Pandit

Thanks to my family members, friends who have
given whole hearted support to write my
first book.

WHY THIS BOOK

Any person, who is familiarized with excel data handling, or who has experienced on using online banking or who has an experience on usage of browser based applications and who need to develop his own application package, this is the first milestone for his learning and development of application. Of course, there is story behind this attempt.

A grocery merchant has approached me and asked whether I can help him to develop accounting application package for his business. I enquired about his education status and he said that he completed his schooling but unable to join the college due to family problems and took over his father's business for their family livelihood. Further he said that he manages the daily business transaction through excel sheet to ascertain his business status. He also said that if he can

able to generate an accounting application package like railway reservation ticket application model, then he can able to train his wife to operate that package and he can concentrate on other new business which he plans as real estate. I am surprised that whether he has any accounting professional knowledge apart from handling excel sheets. I am good at parting the application development knowledge on various platforms and languages but this specific application requires some more knowledge on jargons like debit, credit, transactions, voucher etc. I myself not sure “what to debit and what to credit on any transactions”

Though, I do not know anything about his requirement, I pretend myself that as if I knew and understood his need but still need to know his understandings about his working modalities and enquired about his workings on excel sheet. He very casually said that it is very simple logic and does not require much knowledge to know accounting profession. I was taken a back and surprised to see his expressions but kept quiet. He further narrated as “Expenditures or receipts are grouped under accounting head like cash account, bank account, sales

account, purchase account, party account, expenses account, salary account, rent account etc. etc. As and when I require, I create a reasonable accounting head. Generally account heads are classified under two categories as Capital Nature, Revenue Nature.” His definition made me to sweat just below my throat and flowed through my chest. With a shaky voice, I asked him to give examples of the transactions on capital nature and revenue nature.

Any transactions which have been done for the purpose of day to day routine of the business and have no life after the financial year can be classified as revenue nature and all other transactions can be classified as capital nature. He did not embarrass me and continued that buying goods for sale, paying salary to his workers, receipts from sale of goods etc. are ground under revenue nature and if the transactions pertaining to acquiring land, building, machinery, bank accounts which has life more than a year and for the development / improvisation of business may be classified as capital nature. I am confused about his definition and insisted to narrate about the transactions. He said, “Generally, any

transaction must have minimum five parts, like date of transaction, involved accounting head, brief details about the action, amount involved and the final part very important one as either debit or credit classification. These transactions will be preserved as written document and may be named as voucher. In otherwise, voucher must have minimum of two transactions as debit and credit. I am unable to control myself and asked on higher pitch voice “what does it mean by debit, credit “ Is there any business rule associated on the term of debit / credit.

He thrown his eyeshot on me as “is it worth to explain further”. I lowered my eyes and started exploring something on my palm. He continued, “It is very simple affair sir. Anything goes out of business hall, I treat it as credit type and anything comes inside to the business hall are treated as debit. For example when I buy raw material and paid cash, here goods coming inside the business hence it is debit and cash goes out of business, hence it is credit. One need not know both the debit and credit. Simply, anything goes out is credit and comes in debit and related other accounts are vice versa.

Dear Reader, if you do not understand the business logic mentioned above and some of the accounting concepts, never mind, we shall be able come back while we undertake the real development of application package. We shall also able to get more examples and explanations so as to adopt coding efficiently.

Now, I understood, this young man is very confident on handling any problems related to business logic / rule / frames. Something sparked his inner mind and told him that he made a right choice of approaching me. On seeing the posture of his body language, I also felt comfortable to see him. Suddenly, he presumed that this venture is a joint venture and asked me “Under what platform do you suggest for developing our accounting application package? How long will it take? And how much it may cost? Series of questions fired.

We want to create menu based application like one prevails at railway reservation counter, with which we want to store some data into my computer, like to get back the stored data on the screen and may like to delete the data too. Further, we like to have a report on giving

some information. In nutshell, our application should provide menu based selection, data entry form and formatted report for printing.

You are about to manufacture a product named as financial accounting application and the major raw material used is DATA and you may require various data handling tool. What does it mean by DATA. It is nothing but information / processed information. Also we shall see few of the frequently seen Jargons.

CONTENTS

Table of Contents

<i>1- How to select RIGHT TOOL FOR DATA</i>	14
2 - How to Install Oracle-XE	18
3-HOW TO INSTALL APEX 3.2.1	21
4. HOW TO USE RESOURCED ORACLE-APEX.....	23
5-HOW TO CREATE USER ACCOUNTS	25
6-HOW TO CREATE WORKSPACE	28
7- HOW TO LAND IN APEX HOME PAGE	32
9 – HOW TO CREATE TABLE	43
10-00 HOW TO ALTER TABLE PROPERTIES	55
10.01 How to add another column on the existing table. 55	
10.02 How to change the data type?.....	56
10.03 How to change the column name ?.....	57
10.04 How to delete the column?	58
10.05 How to create another table with the same set of column and data?	59
11 – HOW TO RETRIEVE TEXT DATA	63
11.01 How to display one column from a table	64
11.02 How to display one column data with a label / title	65
11.03 How to display title / label without conversion as uppercase and as we entered.....	65
11.04 How to display all the column values with column name as title	66
11.05 How do we make conditional display of column values?.....	66
11.06 How do we make conditional display with partial given data?	67

11.07	How to use “% “symbol.....	67
11.08	How to use equal (=) operator in condition?	67
11.09	How to use like operator	68
11.10	How to use upper () function	69
11.11	How to use lower () function	69
11.12	How to use Initial () function	69
11.14	How to use LPAD () function	72
11.15	How to use CONCAT function.....	72
11.16	How to use substr() function	72
11.17	How to use LTRIM() function	73
11.18	How to use RTRIM() function.....	74
11.19	How to use TRIM() function	74
11.20	How to use LENGTH() function	75
12 –	HOW TO RETRIEVE NUMERIC DATA.....	77
12.01	How to use numeric operator (+) addition	77
12.02	How to use numeric operator (-) subtraction 78	
12.03	How to use numeric operator (*) multiplication	78
12.04	How to use numeric operator (/) division.	78
12.05	How to use ceil() function	79
12.06	How to use floor() function	79
12.07	How to use power() function	79
12.08	How to use remainder() function.....	80
12.09	How to use round() function.....	80
12.11	How to use max() function	82
12.12	How to use min() function.....	82
12.13	How to use avg() function	83
12.14	How to use count() function	83
13 –	HOW TO RETRIEVE DATE DATA	85

13.01	How to subtract dates to find number of days	85
13.02	How to use add_months() function	86
13.03	How to use months_between() function....	87
14-	HOW TO CONVERT / TRANSFORM DATA	88
	CONVERSION:.....	88
14.01	How to convert date type value to char type value	89
14.02	How to convert numeric data to char type as formatted	90
14.03	How to convert char type data as date type	90
14.04	How to convert formatted char type data as numeric type.....	91
14.05	How to use decode() function.....	93
14.07	How to use group by clause.....	94
14.09	How to use order by clause.....	95
15 –	HOW TO CREATE SEQUENCE	97
16 –	HOW TO CREATE TABLE CONSTRAINT	104
16.01	How to create Primary Key Constraint.....	105
16.02	HOW TO CREATE CHECK CONSTRAINT	107
16.04	HOW TO CREATE FOREIGN KEY	
	CONSTRAINT	111
17 –	HOW TO CREATE TRIGGERS.....	114
18-	HOW TO LAND IN PL/SQL	130
18.01	How to use if conditions	131
18.02	How to use nested if conditions.....	132
18.05	How to use for loop	136
18.06	How to use while loop	136
19-	HOW TO CREATE FUNCTION	138
20-	HOW TO CREATE PROCEDURE.....	141

21 – HOW TO MANIPULATE DATA INSERT / UPDATE / DELETE	153
22 – HOW TO CREATE Application	156
23 – HOW TO CREATE PAGE.....	157
24-HOW TO CREATE PAGE ITEM.....	165
25-HOW TO CREATE LOGO	169
26 HOW TO CREATE TAB / MENU	172
27-HOW TO CREATE BREADCRUMP	179
28-HOW TO CREATE REGIONS	184
29 – HOW TO CREATE BUTTON.....	188
30- HOW TO CREATE LOV	191
31-HOW TO CREATE COMPUTATION	196
32-HOW TO CREATE VALIDATIONS	199
33-HOW TO CREATE PROCESS	202
34- HOW TO CREATE BRANCH.....	204
35-HOW TO CREATE TABULAR-FORM	205
36-HOW TO CREATE MASTER-DETAIL FORM...	211
37 TEST DATA FOR MASTER-DETAIL FORM	218
38-HOW TO CREATE REPORT PAGE	221
39 – HOW TO RUN APPLICATION.....	225

1- How to select RIGHT TOOL FOR DATA

An Oracle Database a collection of data to store and retrieve related information manages a large amount of data in a multi user environment so that many users can concurrently access the same data and prevents unauthorized access. For example when you construct a house, you have different rooms built as kitchen, hall, bed-room, living-room, dining-room, pooja and toilets etc. Entire collection of rooms under the name of house can be called as database and every room can be called as objects and each object serves different nature.

SQL (pronounced SEQUEL) is the programming language that defines and manipulates the database. SQL databases are relational databases, which mean that data is stored in a set of simple relations. An access to living room from kitchen and bed-room has got some relational entry without losing its main objects and identity

SQL*Plus is a tool for entering and running ad-hoc database statements. The communication with which any room identified and access can be called as a tool, here SQL*Plus is another tool of communication to Oracle Database.

DATA may be defined as information, processed information and many attributes can be attached to DATA.

SQL Statements are used to perform all the operations in an oracle database. SQL statement is a group of string, sentence which consists of words. Though it seems to be very simple, but powerful computer program or instruction to access oracle database. SQL statements are divided into four categories: When you enter into the living room, on the right side you can find entry to kitchen and on the left side you shall find an entry to bed room. It is a statement to get access the objects of room in the house database. The same thing can be expressed in different language by my neighbor so as the SQL Statement is nothing but a statement to access Oracle Database.

Oracle Application Express (Oracle APEX) is the right tool for developing our application. The reasons are

- ➔ Oracle Apex is a RAPID APPLICATION DEVELOPMENT (RAD) Tool
- ➔ It has no cost (Oracle Firm has given it as free of cost)
- ➔ Helps to design, develop and deploy the required applications
- ➔ Minimal knowledge alone required on Structured query language and on programming concepts
- ➔ It is Web-browser based applications
- ➔ Do not require software on client machines
- ➔ Oracle Apex is a declarative development tool
- ➔ All Oracle apex development is data driven
- ➔ Developed application will flow from our data design

- ➔ Without additional programming overhead, our applications understand the relationships between tables and other objects created by us
- ➔ Protects your investment
- ➔ Flexibility on migration
- ➔ Securities are not sacrificed

Oracle Application Express runs within the oracle environment. There are two ways of using apex for developing applications.

2 - How to Install Oracle-XE

Oracle Application Express runs within the oracle environment. There are two ways of using apex for developing applications.

1. Download a free copy of Oracle XE (Oracle Database 10g Express Edition) and Apex 3.2.1 and install it in your PC. Installation instructions have information's about handling Oracle Apex software.
2. We can use the hosted version of Oracle Application Express. No installation is required in your machine. Net connectivity alone is required.

HOW TO INSTALL ORACLE XE:

First register yourself in Oracle site (www.oracle.com) using your

Mail id and a password. Goto

www.oracle.com/technology/software/product/database/XE/index.html

Download the Oracle Database 10g Express Edition (Universal) (OracleXEUniv.Exe) and save it in your

desired folder. For example you shall create a folder in D drive as soft source (d:\softsource) and then the downloaded software can be saved in that folder. While downloading, this may ask your user-id and password to know the person's identity of downloaded objects. After saving in the desired location double click the d:\softsource\oracleXEUniv.exe this will start installing the oracle XE to your computer.

- ➔ Click Next on Welcome Page
- ➔ On the License Agreement Window, click Accept Terms and Click Next
- ➔ Choose Destination Location using browse or leave the default and click NEXT
- ➔ Specify the database passwords for system (SYS) database You have to enter the password twice (if you do not have password in mind use the password as admin123 and remember it) Click NEXT

- ➔ On Summary window Click INSTALL
- ➔ On Oracle Database 10g Express Edition
 - Install Wizard Click FINISH

Your destination of oracle home directory gives as d:\oracleXE. Generally the selected port no will be 1521 services 2030 and http listener as 8080 will be displayed. At this stage ignore all these displays.

3-HOW TO INSTALL APEX 3.2.1

Visit <http://apex.oracle.com> using your oracle sign-in password; download the apex_3_2_1.zip to your d:\softsource. After downloading the zip file extract the zip contents in d:\oracleXE folder. Zip file will create directory apex in d:\oracleXE and then the entire zip contents will be extracted. Now go to the desktop find run command box wherein simply type CMD and press enter. You will be landed is DOS window like

```
c:\user\raghu>_   now change the destination as cd
d:\oracleXE\apex and press enter key. You will be
landing in d:\oraclexe\apex>_
```

Now type sqlplus sys/admin123 as sysdba you will get the message under SQL> prompt as connected. In the SQL Prompt Type the following

```
SQL> @apexins SYSAUX SYSAUX TEMP /i/   It will
invoke some of the commands and does something and
will be coming back to SQL prompt. On return again
type in SQL>@apxchpwd ( give the password as
admin123 itself ) If you would have come out of the
```

DOS prompt, again got dos window using cmd command and change directory to d:\oracleXE\apex.

```
D:\oraclexe\APEX> sqlplus sys / admin123 as sysdba
```

```
SQL > apxldimg.sql d:\oracleXE
```

```
SQL > apxxepwd.sql admin123
```

```
SQL > Quit
```

```
D:\OracleXE\Apex >Exit
```

Now that, your installation of OracleXE, Apex is over. If you find any problem on installing these software's, it is better to consult some people who are in the line to read out the installation procedures laid in oracle site itself. Generally, this is simplest installation procedure and you will not face any problem on installation procedures.

4. HOW TO USE RESOURCED ORACLE-APEX

Alternatively, we shall see how to use web resourced Oracle APEX:

- ⇒ Go to <http://apex.oracle.com>
- ⇒ Click on the Sign up for Account
- ⇒ Click Next on welcome page
- ⇒ Type your name, email address and click Next in Identifying the person as administrator of hosted Oracle Apex
- ⇒ Type workspace name and click NEXT
- ⇒ If the name of the workspace is not unique, system will prompt you to enter another workspace name and after entering the revised name click NEXT
- ⇒ System will ask schema name wherein you can store your tables or objects etc. Generally, workspace name itself will be given as schema name too. Leave the default initial space allocation of 10MBs and click NEXT

- ⇒ System will require the reason for using hosted service. Type research and develop new application and then click NEXT
- ⇒ In some systems, system will generate verification code and you will be asked to type activation code in the system so that hosted service will understand that this creation of account has been done by human being instead of automated programs.
- ⇒ You will receive credentials through email.
- ⇒ On receipt of email, double click the URL will take you to Oracle APEX login environment.
- ⇒ Every time, you give the workspace name, schema name and password and then you are landed in Oracle APEX environment. Entire ground is ready for your play.

Database Connection By default, the installer configures your operating system to start database automatically.

5-HOW TO CREATE USER ACCOUNTS

One must have domain or registration to use the oracle-APEX. Generally this will be called as user accounts. Users' profile, rights, roles will be defined. User may be declared as an administrator who has lots of power to access, to develop application with some restricted access and a user to use the created application without modification or correction rights.

Connect to ORACLE with SQL*PLUS: Two ways of starting SQL*PLUS

1. Start →

All Programs →

Oracle Database 10g Express Edition

→

Run SQL Command

Windows will PopUp with the following message:

SQL*Plus: Release 10.2.0.1.0 – Production on Wed Jun
2 03:07:00 2010

Copyright © 1982, 2005, Oracle. All rights reserved.

SQL>_

Under SQL Prompt type connect system press <enter>
system will prompt with password ... Type admin123
and Press <enter>

Connected message will be displayed along with SQL>_
prompt.

At present we will type EXIT and return back.

2. Type sqlplus in Command Prompt Window and
press <enter> will also take you to SQL Prompt
and rest of the above command may be tried.

We shall create a user under SQL Prompt with some of
the rights.

Login SQL*Plus connect with username as system and
password as admin123 and get connected.

Now, we are going to create a user and the name
of the user-id is learner. His password also we use the
same user-id defined. In this case learner is a user-id and
learner is the password too.

```
SQL> create user learner identified by learner;
```

press <enter>

User created message will be displayed along with SQL
prompt.

Now our next task is assigning rights, since we like to register the user learner as administrator, developer and application user, we are authorizing by minimal three rights as connect, privilege to get connected with database, and to use all the resources of oracle-APEX and as database administrator. Since database administrator can create many user with restricted rights of developing using the application.

Now type in the sql prompt as

```
SQL> Grant connect, resource, dba to learner;
```

Grant succeeded message will be displayed.

Type Exit and come out of SQL prompt.

Meaning of these commands and usage will be seen on later part. Temporarily you understand that you have created user as learner and the password is learner.

Now we shall enter into Apex environment:

6-HOW TO CREATE WORKSPACE

Generally, when a user needs to do some actions, primarily, he need to be allotted workspace and it should be named. A single workspace can be connected with many user-id or schema or accounts. For example, accounts are the workspace which can be operable by many schemas, receivables, payables, inventories, ledgers. Further workspace can be initiated or created by system administrator only. Hence at the time installing the oracle-APEX we would have given the system-administrator name, schema, password etc. In our case, internal is the system-administrator, admin is the schema operable this administrator and admin123 is the password to get into the oracle-APEX as administrator.

Start →

All Programs →

Oracle Database 10g Express Edition

→

GoTo Database Homepage

Now your browser will start with the following URLs:

<http://127.0.0.1:8080/apex/f?p=4550:1:xxxxxxxxxxxxxxxxxx>
[XXXXX](#)

Enter Application Express workspace and credentials.

Workspace

UserName

Password

Click <LOGIN> you will land in Login Page wherein

→ Type internal against workspace

→ Type admin against username

→ Admin123 against password or the password
given at the time of creation of apex

Once we logged in as administrator, our next task is creating workspace for learner. Since, we are at the learning stage of creating application; we shall assign the workspace name as learner itself. Here this learner workspace and the user itself as administrator, we shall create workspace as learner, schema as learner and administrator password as learner.

On successful login, your screen will flash with four Icons as

- ==> **MANAGE SERVICE**
- ==> **MANAGE WORKSPACE**
- ==> **MANAGE APPLICATIONS**
- ==> **MONITOR ACTIVITY**

Select “**MANAGE WORKSPACE**”

→ create “**WORKSPACE**” from the sub menu:

As soon as you land in Identify Workspace window

- Type learner against workspace name
- Some blah blah blah against Workspace description
- **NEXT**

Land in Identify Schema Window

- **YES** against Re-use existing schema
- select the schema as **LEARNER** (hope

You have not missed the chapter how to
create

User accounts.) → NEXT

Land in Identify Administrator Window

- Type Learner as Administrator Username
- Type learner as administrator password
- Your choice or leave against first name,

last name

- give some email address against Email
- NEXT

Land in Confirm Request Window

- Verify the contents → CREATE

You will be prompted that Workspace
successfully created and workspace learner and
administrator learner. Simply Click LOGOUT..

7- HOW TO LAND IN APEX HOME PAGE

Click Login and enter learner against workspace, learner against schema and password as learner and click LOGIN

On successful Login Oracle Application Express will flash three ICONs as follows?

==> APPLICATION BUILDER

==> SQL WORKSHOP

==> UTILITIES

Application Builder → Acts as interface between database objects like tables, procedures and Helps on creating application, which in turn has objects like pages, regions, buttons, items, validations, computations, processes etc.

SQL Workshop: provides tools to manage, view database objects using web browser concepts.

→ **OBJECT BROWSER:** View, create, modify, browse Database objects like table, view,

sequences, Procedures, triggers, functions, compiling, drop etc.

→ **SQL COMMANDS:** run sql commands, Anonymous PL/SQL blocks, scripts, and saved queries

→ **QUERY BUILDER:** Using graphical interface, search, View, modify, establishing relationships and fiddling with Database objects.

Utility: Load / Unload Data, generating DDL export import

Lots and Lots of Jargons. Just ignore. On seeing the icons on Apex Home Page and sub Menu Icons decryption alone has been narrated here. Even if you do not understand right now, just ignore and proceed you will have very bright understanding after creating a model application. While you are learning car driving, instructor use to tell this is brake, accelerator, ignition, and some rules. I know remembering on the very first session and understanding is a tough task. After learning, did you ever remember the definition given on car objects? Just satisfy yourself you are able to enter into Oracle Apex Home Page.

Now, we slightly change our self from routine and try to learn some of the SQL Commands which is a brain nerve. From now on our learning process of SQL starts with SQL Workshop.

Structured Query Language (SQL) pronounced as sequel is a simple command language used by database administrators, application developers and users of applications.

8- HOW TO CONVERT EXCELL DATA AS ORACLE TABLE DATA

As you are aware Oracle Application Express swings around Oracle Database and in a way it can be classified under CLASSIC RELATIONAL DATABASE. In any relational database groups of data are stored / implemented as tables. Now we shall try to understand this table concept differently. At the start of the session, you said that you are having excel sheets wherein account heads are stored as follows

In excel sheet, Column A is used for Account_Head and Column B is used for Account Type.

Account Head	Account Type
Capital Account	Capital
Cash Account	Capital
Bank Account	Capital
Fixed Asset	Capital
Loan Account	Capital
Inventory	Capital
Reserves and Surplus	Capital
Investment	Capital
Purchase	Revenue

Sales	Revenue
Salary	Revenue
Expenses	Revenue
Stock on hand	Revenue
Discounts	Revenue
Payables	Capital
Receivables	Capital
Sundry Debtors-Ms.X Ltd	Capital
Sundry Creditors-YYY Ltd	Capital
Discounts	Revenue
Depreciation	Revenue
Interest	Revenue

Let us equate this excel sheet with a table structure.

Table Name is Account_Head_Master (Sheet name).

Column Names are Account_head, Account_type. The vertical row numbers can be classified as row number, record id. The data's like Purchases, sales, capital, revenues are data's.

Now we shall try automatic table creation and population directly from the above excel sheet in Oracle Apex environment.

Importing Data's from Excel and creation of table and storing steps are:

- ⇒ Log into Oracle Apex with the given workspace name, schema name and the password (learner / learner / learner)
- ⇒ Click **UTILITIES**

On clicking Utilities, new page will get displayed with following Icons.

- ⇒ **DATA LOAD / UNLOAD**
- ⇒ **GENERATE DDL**
- ⇒ **OBJECT REPORTS**
- ⇒ **RECYCLE BIN**
- ⇒ **DATABASE MONITOR**
- ⇒ **APEX VIEWS**
- ⇒ **SCHEMA COMPARISON**
- ⇒ **ABOUT DATABASE**

From the above eight Icons select Data Load / Unload

- ⇒ Click **Data Load / Unload** (cascade pictured)
- ⇒ Click Load
- ⇒ Click Load Spreadsheet Data
- ⇒ Select New Table under “Load To” and select Upload file from “Load From” CLICK NEXT
- ⇒ Browse and select the .csv file and CLICK NEXT
- ⇒ Type “ACCOUNT_HEAD_MASTER” against table name prompt and click NEXT
- ⇒ Select “Create New Column” under Primary Key From, Type “REC_ID” against New Primary key column, Leave the default PK constraint name, select “Generate From New Sequence” from Primary key population and type REC_ID_SEQ against sequence and CLICK “LOAD DATA”

You created the table successfully. Congrats. Now we shall play with created accounts_head_master table and understand some of the sql statements behavior.

- ➔ Login to **APEX** using workspace / schema / password
- ➔ Click **SQL Workshop**
- ➔ Click **SQL Command**

This will display a new page and it may be called as SQL Command Editor. The SQL command editor horizontally split into two portions. Upper portion is used for giving SQL Command and the bottom portion will display the results derived out of SQL Statements / commands.

Having created a table from excel data, if we want to see the structure of the table and the contents, give command as DESCRIBE << table name >>. In our case we have already created Account_Head_Master, hence

Describe account_head_master

Or

desc account_head_master

may be given SQL Prompt or in the SQL Command window. If we use this command in SQL Command Window, our sql commands need not be ended with semi-colon (;) otherwise generally sql command should

be terminated with semi-colon (;) From the above said, we shall be able get the result as

Table Name : Account_head_Master

Column Name	Data Type	Length	Precision
Rec_id	Number		
Account_head	Varchar2	30	
Account_type	Varchar2	30	

Since, we converted the excel data as table, we are able to understand the table name, column name, data with respect to stored data in the excel sheet. But here we are seeing the new word as Data Type. One may understand that there are basically two main data types are prevailing in the Database as Numeric, Non-Numeric. Further anything which is subject to arithmetical operations can be classified as numeric data and other vice-versa data are classified as Non-Numeric. But Here there are two data types as Number, Varchar2 has been used. Let us try to understand the definition of these data types. Character data types store alphanumeric data, text.

CHAR Data type

The `CHAR` data type specifies a fixed-length character string. Oracle blank-pads the value to column length when data length shorter than declared.

VARCHAR2 Data type

The `VARCHAR2` data type specifies a variable-length character string. Oracle stores each value in the column exactly as we enter.

NUMBER Data type

The `NUMBER` data type stores any numeric values. Default length will be 22 bytes. `NUMBER (p, s)` where `s` means no of digits after decimal and `p` intend to specify the no of digits before decimal.

DATE Data type

The `DATE` data type stores date and time information. `DATE` value, Oracle stores the following information: century, year, month, date, hour, minute, and second.

Using excel sheet we have created a table for testing purpose, whereas, we intend to create new

application for which different tables are needed and are going to follow in various chapters.

9 – HOW TO CREATE TABLE

We have an intention of creating two application software's using this book content, so that one can understand the barest minimum concepts, and want to gain the confidence booster. One application will be handling voucher entries and result can be viewed through profit and loss account, balance sheet. Other application to understand some of the concepts in different approach. With this intention, required tables and business rules, process logic, validation, computation etc. will be learnt.

The following tables require for our application development. Table design will be discussed and associated business rules will be discussed on later part of the book.

I. Control-Master where in voucher-sequence number, Ledger-code sequence number, which has got relation with other tables are controlled and maintained.

Table Name : RAAK_CTRL_MASTER (It

maintains only one record and frequently gets updated as the last sequence number to be assigned on requirement)		
Column_Name	Data Type / Width	Remarks
Record_Id	Number	If length is not specified default value will be taken as 22. This column will be populated from the system generated sequence number
Ledger_seq	Number (16, 0)	Length will be 16 and decimal portion will be 2 digits. This sequence will be used to generate Ledger-code with some concatenation of other values.
Voucher_seq	Number	This number will be

	(16,0)	used in the voucher Master as Voucher Number and after allocation, this voucher_seq will be updated with increment of 1 (One)
--	---------	---

II. Ledger-Master will accommodate Ledger-Code, Ledger-Description, and Ledger-Type which will be used in the voucher transactions.

Table Name : RAAK_LEDGER_MASTER		
Column Name	Data Type/ Width	Remarks
Record_Id	Number	This column will be populated from the system generated sequence number
Ledger-Code	CHAR (10)	Length will be 10 if

		<p>data is not provided for full-length then spaces will be stored. Frist Four digits will be taken from raak_ctrl_Master (Ledger_seq) will be padded with zero; next three characters will tell whether BS (balance sheet item or PL Profit and Loss account item) and remaining three digits will be the sequence of display. This column must be Unique. For example 'Balance sheet' report and Cash and Bank Balances Ledger Code may be</p>
--	--	--

		termed as '0001-BS-00' against Ledger_Code.
Ledger_Name	varchar2(200)	The maximum length will be 200 and real data length alone will be counted as storage bytes. This column must be unique
Report_Type	Char (2)	'BS' will be treated as balance sheet report item 'PL' will be treated as Profit and Loss Report item

III. Transaction_Master wherein the transactions total, voucher number will be stored

Table Name : RAAK_TRANS_MASTER		
Column_Name	Data Type / Width	Remarks
Record_Id	Number	
Voucher_Type	Char (1)	'R' – Receipt, 'P'- Payment 'J'- Journal
Voucher_Number	Char (10)	This column linked with transaction Master (Voucher Number) Ex: 'RPT-000001'
Voucher_Date	Date	Application date format need to be maintained as 'DD-MM-YYYY'
Voucher_naration	Varchar2 (400)	Description about the transactions
Debit_Total	Number(16, 2)	Calculates the total of debits exists in

		details record with respect to this voucher number
Credit_Total	Number(16, 2)	Instead of debit, credit applies.

IV. Transaction_Details will accommodate nature of transactions with the link to transaction_master through voucher number.

Table Name : RAAK_TRANS_DETAILS		
Column_Name	Data Type / Width	Remarks
Record_Id	Number	
Voucher_Number	Char (10)	This column linked with transaction Master (Voucher Number)
Ledger_Code	Char (10)	This column linked with Ledger Master (Ledger Code)

Voucher_naration	Varchar2 (400)	Description about the transactions
Debit	Number(16, 2)	
Credit	Number(16, 2)	

Having designed the four tables, we will borrow or copy the SQL scripts for creating the tables

- login Apex Home Page
- SQL Workshop
- SQL Command

```
Create Table raak_ctrl_master (
    Record_id    number,
    Ledger_seq   number (16, 0),
    Voucher_seq  number (16, 0))
```

create is the key command word for creating any object in the database. Out of various objects table is one object wherein the columns are defined with data type and width. Hence create table should be followed with << table_name >> and the columns to be included in this table should be embedded with parenthesis. On giving this script in the command window, sql statement will

run and 'table created' display will be in the result window. Likewise the other three tables can be created.

Apex Home → SQL Workshop → SQL Commands
Window

Type

```
Create table raak_ledger_master (  
    record_id number,  
    ledger_code char (10),  
    ledger_name varchar2(200),  
    report_type char(2))          CLICK RUN
```

In the result Window, Table created Message will get displayed.

Apex Home → SQL Workshop → SQL Commands
Window

```
create Table raak_trans_master  
( Record_id   number,  
    voucher_type char(1),  
    voucher_number char(10),  
    voucher_date date,  
    voucher_naration varchar2(400),
```

```
debit_total number(16,2),  
credit_total number(16,2))      CLICK RUN
```

In the result Window, Table created Message will get displayed.

Apex Home → SQL Workshop → SQL Commands
Window

```
create Table raak_trans_detail  
( Record_id   number,  
  voucher_number char(10),  
  ledger_code   char(10),  
  voucher_naration varchar2(400),  
  debit number(16,2),  
  credit number(16,2))      CLICK RUN
```

In the result Window, Table created Message will get displayed.

The following table creation scripts and the procedure need to be executed before creation of report page. The proposed Balance_sheet, profit_loss table is temporary table wherein the data's will be pooled as required through some procedure before print / display. Generally,

whenever, we require to initiate reports, these tables data will be removed, and the data's from voucher_details will be accumulated and totaled by ledger_name and will be stored in these tables. In otherwise, Ledger_names wise, the data's will be summed and if ledger_name belongs to balance sheet then it will be inserted in balance_sheet table and if the report_type of ledger_name belongs to profit and loss then it will be inserted in profit_loss table.

Further, credits total will be placed in liability / income and all debit totals will be stored against asset / expenditure.

```
CREATE TABLE "BALANCE_SHEET"  
(  
    "RECORD_ID" NUMBER,  
    "LEDGER_NAME" VARCHAR2(200),  
    "LIABILITY" NUMBER(16,2),  
    "ASSET" NUMBER(16,2),  
    CONSTRAINT "BALANCE_SHEET_PK" PRIMARY KEY  
    ("RECORD_ID") ENABLE  
)
```

```
CREATE TABLE "PROFIT_LOSS"
```

```

(      "RECORD_ID" NUMBER,
      "LEDGER_HEAD" VARCHAR2(200),
      "EXPENDITURE" NUMBER(16,2),
      "INCOME" NUMBER(16,2),
      CONSTRAINT "PROFIT_LOSS_PK" PRIMARY KEY
("RECORD_ID") ENABLE
)
/

```

10-00 HOW TO ALTER TABLE PROPERTIES

At the time of creation or on change of system design one may require to rectify the table properties.

=>Home → SQL Workshop →

=>**create a test_table**

=> “create table test_table (test_col number (16, 2))”

10.01 How to add another column on the existing table.

In the above created table test_Table has got only one column and want to add another column test_col2 as below:

=>ADD one column as test_col2 with a data type char (10)

Through object Browser	Through sql command Window
→ Tables	Alter table test_table

→ TEST_TABLE	add (test_col2
→ ADD Column	char(10))
→ Test_col2 against add column	
→ CHAR against Type → 10 against Length NEXT	
→ Finish	

10.02 How to change the data type?

The data type which has been already created as char(2), and need to be changed as number with width and precision.

MODIFY the column test_col2 char (10) as Test_col2 number (16, 2)

Through object Browser	Through sql command Window
→ Tables	alter table testtable

→ TEST_TABLE → Modify Column → Test_col2 against column → NUMBER against Type → 16 against Length Precision → 2 against scale → NEXT FINISH	modify (test_col2 number(16,2))
---	------------------------------------

10.03 How to change the column name ?

=> RENAME the column Test_col as Test_col1

Through object Browser	Through sql command Window
→ Tables → TEST_TABLE → RENAME Column → Test_col	alter table test_table rename column test_col to test_col1

against current column → Test_col1 against New Column Name → 16 against Length Precision → NEXT FINISH	
--	--

10.04 How to delete the column?

Assumption has been made that no data exists in the "to be deleted column".

DROP table column test_col2 in test_table

Through object Browser	Through sql command Window
→ Tables → TEST_TABLE → DROP Column	alter table testtable drop column test_col2

<p>→ Test_col2</p> <p>against remove column</p> <p>→ NEXT FINISH</p>	
--	--

10.05 How to create another table with the same set of column and data?

COPY table test_table_new to test_table_old

Through object Browser	Through sql command Window
<p>→ Tables</p> <p>→ TEST_TABLE_NEW</p> <p>→COPY</p> <p>→ test_table_old in new table name</p> <p>→ select all columns in columns</p> <p>→ select all rows against display</p>	<p>create table test_Table_old as</p> <p>(select * from test_Table_new)</p> <p>where will the option to copy selected rows, specific columns instead of * restricting columns</p>

<p>→ NEXT FINISH</p>	<p>Copy a table into another table (with all columns, with all rows)</p> <pre>create table <<table_name>> as select * from <<to_be_copied file_name>> copying selected columns from another table create table <<new_table_name>> as (select col_1, col_5, col_2 from <<old_table>></pre>
----------------------	--

→ Tables	create table test_Table_old
→ TEST_TABLE_NEW	as
→COPY	(select * from
→ test_table_old in new	test_Table_new)
table name	where will the option to
→ select all columns in	copy selected rows,
columns	specific columns instead
→ select all rows against	of * restricting columns
display → NEXT	Copy a table into another
FINISH	table (with all columns,
	with all rows)
	create table
	<<table_name>> as
	select * from
	<<to_be_copied
	file_name>>
	copying selected columns
	from another table create
	table
	<<new_table_name>> as
	(select col_1, col_5,

	col_2 from <<old_table>>
--	--------------------------

11 – HOW TO RETRIEVE TEXT DATA

Oracle provides one default table and named as DUAL. It has got only one column as DUMMY and data type is varchar2 (1). This table will be used to handle memory variables, temporary calculations and non-stored data of regular tables. Developers will use this table for testing much syntax. We shall try to use this table for testing the formatting, retrieving, various text data conversions and few of the Oracle functions etc.

Let us assume there are two data such as ‘cash account’ and ‘sales account’ are stored in the table of LEDGER_MASTER under the column of Ledger_name. How it can be stored in the table will be seen in the later chapter.

“select” is the key word to be used for picking data’s as “create command” a followed with <<column_names>> (the data needed against column names) and the location need to be specified as “From” and the database object. The syntax is

```
“select      <<column_names>>      from  
<<table_name>>;
```

If we want to see the data available in ledger_name of Ledger_Master table, then we shall give the syntax as “select ledger_name from ledger_master;” and this will retrieve data and display as follows:

11.01 How to display one column from a table

SYNTAX:

```
select ledger_name from ledger_master;
```

RESULT:

LEDGER_NAME

cash account

sales account

Generally, the column title or Heading label will be derived from the column_name itself. If we need the Title / column_heading differently we shall add the “as ledger” after column_name but before “from”.

11.02 How to display one column data with a label / title

SYNTAX: select ledger_name as ledger from
ledger_master;

RESULT:

LEDGER

cash account

sales account

11.03 How to display title / label without conversion as uppercase and as we entered

Here an interesting aspect, we gave the title “ledger” in lower case letters, where as it displayed the title in upper case letters. If we want to get the title as we entered / typed then the title should be embedded by double-quotes.

SYNTAX: select ledger_name “ledger” from
ledger_master;

RESULT:

ledger

cash account

sales account

11.04 How to display all the column values with column name as title

Before, we see some more conditional retrieval and usage of functions; let us see what are the records available in the raak_ledger_master. If we want to retrieve all the columns available in the table may be referred as “*” instead of giving all the column_name with a separator comma.

SYNTAX: select * from raak_ledger_master;

RESULT: All the columns will be get displayed along column_name with a heading

11.05 How do we make conditional display of column values?

We like to retrieve the entire record for the value of “cash account” available in the table.

SYNTAX: Select * from raak_ledger_master where ledger_name='cash account';

RESULT: Oracle produces the relevant required data.

Here, we have added the clause as “where” and followed with column_name to be referred / checked in the table. “=” sign has been used for comparison / evaluation.

11.06 How do we make conditional display with partial given data?

11.07 How to use “% “symbol

11.08 How to use equal (=) operator in condition?

We had a vague memory that portion of the data can be given for comparison / evaluation; the left data portion may be replaced by “% “symbol.

SYNTAX: Select * from raak_ledger_master where ledger_name='cash%';

RESULT: no data found

It is surprised that we have data but it does not return the value. We just made one mistake. Our column_name ledger_name is varchar2 and we have to use ‘LIKE’ operator instead of “= “symbol.

11.09 How to use like operator

SYNTAX: Select * from raak_ledger_master where ledger_name like 'cash%';

RESULT:

RECORD_ID	LEDIGER_CODE	LEDGER_NAME	REPORT_TYPE
1	BS-0001-00	CASH ACCOUNT	BS

Hence we shall understand “LIKE” is another string comparison operator as equal sign exists.

A function in oracle behaves and produces many results in a way pre-written codes accepting few parameters and producing one return values. Every function has got sets of parenthesis where in user pass the parameters / options required. The functions which are handling string, character type of data, produces two types of the results. String Function may change the original text information / data, or tells about the properties of the data. Some functions will convert all the text characters in terms of upper case letters and some functions may produce length of the data, different form of presentation etc. etc.

LOWER () → converts every letter in a string to lower case,

UPPER () → converts every letter in a string to upper case,

INITCAP () → Initial capital. Capitalize the first letter of a word or series of word.

11.10 How to use upper () function

11.11 How to use lower () function

11.12 How to use Initial () function

Options like column_name may be given within parenthesis or the letters / string embedded by single quotes will give desired result. For example

SYNTAX: select upper (ledger_name) "UPPER",
InitCap (ledger_name) "Init Cap",
lower ('ORACLE APEX') from raak_ledger_master;

RESULT:

UPPER

INIT CAP

LOWER

CASH ACCOUNT	Cash Account	oracle apex
--------------	--------------	-------------

SALES ACCOUNT	Sales Account	oracle-apex
---------------	---------------	-------------

On seeing the result, you shall find syntax is self-explanatory and achieved the desired results.

11.13 How to use concatenate

Concatenation (||) -- two vertical lines -- pipe symbol can be used to combine or join two string, character type column name. In case, if one need to combine numeric data to string, then numeric data need to be converted as string or character type data and then join / combine two different string data. We will try to evaluate using sql statement where both concatenation function and pipes usage.

SYNTAX: select concat ('Miss ', 'Anugraha') as "FUNCTION_USE",

'Miss ' || 'Anugraha' as "SYMBAL_USE" from dual;

RESULT:

FUNCTION_USE	SYMBAL_USE
--------------	------------

Miss Anugraha	Miss Anugraha
---------------	---------------

We have one requirement as voucher no has been declared as character and length 10 char (10). The first three characters should be 'RPT', 'PYT', 'JOU' and

should be followed some sequence number. The number should have leading zeros.

For example Receipt voucher number 1027 should be stored as RPT-001027. To achieve this we are going use two functions as combined functions. Everyone can understand through book, the actions desired by function command but in real time application environment, many time warrants combining two or more functions. In this case, we are going to use **LPAD ()**, **CONCAT ()**. LPAD allows you to “pad” on the left side of the column with set of characters. The characters can be spaces, periods, commas, letters, numbers, and sign. To do this, we have to give three parameters.

- ➔string or column_name

- ➔Total length required on result string

- ➔character need to be defined (to be padded characters)

LPAD (string, length, to be padded character)

CONCAT (string1, string2) or CONCAT (string1, column_name) or

CONCAT (column_name1, column_name2)

11.14 How to use LPAD () function

11.15 How to use CONCAT function

SYNTAX: select concat ('RPT', LPAD (1027, 6, '0'))
"Voucher_number" from dual;

RESULT:

VOUCHER_NUMBER
RPT001027

Instead of left side padding if required right side padding, we may use RPAD command.

11.16 How to use substr() function

Another frequently used function and we may require in our proposed application development will be SUBSTR() called as substring function. This function also has three parameter options

- ➔string or column_name
- ➔starting position

➔number of characters needed from the starting position

SUBSTR ((string, start, count). Assume RPT-001027 is a string; I want to pick out 1027. One is located in 7th position and followed 4 characters (count) needed.

SYNTAX: select SUBSTR ('RPT-001027',7,4) AS RETURN_VALUE FROM DUAL;

RESULT :

RETURN_VALUE
1027

11.17 How to use LTRIM() function

When we need to remove left most unwanted or to be eliminated characters on the column we need to use the following syntax:

Syntax: select ltrim('aaa,bbb','') as result from dual;

Result :

Result
aaa,bbb,

11.18 **How to use RTRIM() function**

When we need to remove right most unwanted or to be eliminated characters on the column we need to use the following syntax:

Syntax: `select rtrim (' , aaa, bbb',' ,') as result from dual;`

Result:

```
Result
-----
,aaa,bbb
```

11.19 **How to use TRIM() function**

When we need to remove both sides right and left most unwanted or to be eliminated characters on the column we need to use the following syntax:

Syntax: `select trim(',') from ' ,aaa,bbb, ' as result from dual;`

Result :

```
Result
-----
```

aaa,bbb

11.20 How to use LENGTH() function

To find out the no of characters in any column or text data, we need to use the following syntax:

Syntax: select length('aaa,bbb,') as result from dual;

Result :

Result
9

11.21 How to use INSTR() function

To find out the position of required characters on the occurrence in any column or text data, we need to use the following syntax:

instr(column_name/data , to be located characters set, start from , occurrence)

instr('aaa,bbb,' , ',' , 1 , 2)

Syntax: select instr('aaa,bbb,',',', 1,2) as result from dual;

Result :

Result

5

11.22 How to use REPLACE() function

We want to replace all the commas in data as dash
replace(column_name/data , to be replaced characters
set, replace with characters)

replace('aaa,bbb,' , ',' , '-')

Syntax: select instr('aaa,bbb,' , ',' , '-') as result
from dual;

Result :

Result

-aaa-bbb-

12 – HOW TO RETRIEVE NUMERIC DATA

Apart from numeric static value, oracle classified some functions under three categories like single value function, group of values, lists of values. All parameters options involved in string function like option, pair of parentheses.

Single Value Function : Using this function one can select one column of the one row out of whole table. In another way, after considering many values and generate a single value.

12.01 How to use numeric operator (+) addition

SYNTAX: select (8+3) "add" from dual ;

RESULT:

add

11

12.02 How to use numeric operator (-) subtraction

SYNTAX: select (8-3) "subtract" from dual ;

RESULT:

subtract

5

12.03 How to use numeric operator (*) multiplication

SYNTAX: select (8*3) "multiply" from dual ;

RESULT:

multiply

24

12.04 How to use numeric operator (/) division

SYNTAX: select (8/3) "divide" from dual ;

RESULT:

divide

2.66666

12.05 How to use ceil() function

Syntax: select ceil(1.2) "ceil" from dual ;

RESULT:

ceil

2

12.06 How to use floor() function

Syntax: select floor(1.2) "floor" from dual ;

RESULT:

floor

1

12.07 How to use power() function

Syntax: select power(8,3) "power" from dual ;

RESULT:

power

512

12.08 How to use remainder() function

Syntax: remainder(8,3) "remainder" from dual ;

It gives the remaining short value to make 8 as divisible

RESULT:

reminder

-1

12.09 How to use round() function

Syntax: select round(123456.89,-2) "round_2_100" from
dual ;

RESULT:

round_2_100

123500

In this single value function, you can note down under
round function, -2 indicates rounded to hundred, -1

indicates rounded to tens, 0 rounded to rupee, 1 rounded to ten paisa's

12.10 How to use mod() function

SYNTAX: select mod(8 , 3) "modulus" from dual ;

RESULT:

modulus

2

Though, there were many functions are available, we are concentrating as recap and some very important functions frequently used and likely to be used in our proposed application development.

Now our next task is to analyses some of the Aggregate functions.

Aggregate Functions: These functions mainly forming part of statistical nature since, it considers all the rows of a table and pick out one value among suiting to the function nature. For example, day wise rain-fall has been maintained in a table and “select max(rainfall) “max”, min(rainfall) “min”, avg(rainfall) “avg”,

Count(*) “no.of records” from rainfall_master” will produce the required values as maximum rainfall, minimum rainfall, average rainfall, no.of records exist etc. Assume three records rainfall as 23.7, 31.3, and 12.0

12.11 How to use max() function

```
select max(rainfall) “max” from rainfall_master
max
- - -
31.3
```

12.12 How to use min() function

```
select min(rainfall) “min” from rainfall_master
min
- - -
32.0
```

12.13 **How to use avg() function**

```
select avg(rainfall) "avg" from rainfall_master
```

```
avg
```

```
---
```

```
22.33333
```

12.14 **How to use count() function**

```
select Count(rainfall) "no.of records" from  
rainfall_master
```

```
no.of records
```

```
---
```

```
3
```

List Functions: In one particular row, various values are stored as morning_price, noon_price, evening_price date wise.

If we want to know the least or greatest value among rows, we can provide syntax as “select date_of_trans “date”, greatest(morning_price, noon_price, evening_price) “Gold High Price”, least(morning_price,

noon_price, evening_price) “Gold low Price” from
gold_price_master ;

13 – HOW TO RETRIEVE DATE DATA

Date is another oracle data type like any other type of varchar2, char, number but it has its own unique properties. Sysdate does not require any parameters or options will return the value of your system's current date and time.

Date arithmetic consists of addition and subtraction only. If we subtract one date from another date, it will return the value as number of days between these two dates. Likewise, if you add number of days in a given date it will return after adding no.of days given for calculation. Likewise, no. of months also can be added to a date or deduct / subtract no.of months from a given date. Let us see few of the date arithmetic examples:

13.01 How to subtract dates to find number of days

⇒ Number of days between sysdate / current
date and 05-mar-1991

SYNTAX: select (sysdate - to_date('05-mar-1991'))

"No.of Days" From dual ;

RESULT : No.of Days

7043.9634490740740740740740740707407407

13.02 How to use add_months() function

⇒ Retirement date by adding 12 * 60 months
from 05-mar-1991

SYNTAX: select add_months(to_date('05-mar-
1991'),(12*60)) "Retirement date" From dual ;

RESULT :

Retirement_date

05-Mar-51

⇒ Finding Birth day from Retirement date by
subtracting 12 * 60 months from 05-mar-
2010

SYNTAX: select add_months(to_date('05-mar-
2010'),-(12*60)) "Birth Date" From dual ;

RESULT :

Birth Date

- - - - -

05-Mar-60

13.03 How to use months_between() function

⇒ Finding Number of months between two
dates 05-mar-2010 and 05-mar-1991

SYNTAX: select months_between(
to_date('05-mar-2010'),to_date('05-mar-1991'))

"No.of Months" From dual ;

RESULT : No.of Months

228

Usage of Extract function :

13.04 How to extract year from date column

SYNTAX:

select extract(YEAR from sysdate) "YEAR",
extract(month from sysdate) "MONTH",
extract(DAY from sysdate) "DAY" from dual ;

RESULT :

Year Month Day

2010 6 16

14- HOW TO CONVERT / TRANSFORM DATA

CONVERSION:

→To_char Transforms a DATE or NUMBER into a character string.

→To_date Transforms a Number, char or varchar2 into a DATE

→To_Number Transforms a char, varchar2 into a NUMBER.

One may enter the date in different ways such as
01-dec-2010, 01-12-2010, 2010-12-01, 12-01-2010

SQL EXPRESSION	RESULT
TO_DATE('01-DEC-2010')	'01-DEC-10
TO_DATE('01-12-2010','DD-MM-YYYY')	

We have seen some of the retrieval on selective data types like char, varchar2, number and data. Of course, the seen examples are very illustrative and not

exhaustive. The aims of the said examples are with a minimal knowledge or recap of your oracle sql strength, we should be able to generate application software. Apart from selective data type retrieval, there are certain functions are used for conversion of data types

14.01 **How to convert date type value to char type value**

1. To_char() Transforms a date or number into a character string. For example, sysdate generally displays date as dd-mon-yy (18-Jun-10). In case, if we want to see the date in dd-mm-yy format, we shall use the “*select to_char(sysdate,'dd-mm-yyyy') as formatted_date from dual*” and result will be

Formatted_Date

18-06-2010

14.02 **How to convert numeric data to char type as formatted**

If we want to see the numerals, in comma separated format, we shall use the “*select to_char(123456.89,'99g99g999d99')*as *formatted_date from dual*” or “*select to_char(123456.89,'99,99,999.99')*as *formatted_date from dual*”. In either case the result will be

FORMATTED_NUMBER

1,23,456.89

14.03 **How to convert char type data as date type**

To_Date () Transforms a number, char or varchar2 into a date, for example user enters the date as ‘1961-08-25’ in the yyyy-mm-dd format referring 25th august 1961. We need to convert this data into date format

Syntax: select to_date('1961-08-25','yyyy-mm-dd')as chr_2_date from dual

Result:

CHR_2_DATE

26-AUG-61

14.04 **How to convert formatted char type data as numeric type**

3. To_number() Transforms a char or varchar2 into a number. For example if we want to add the two formatted numbers 1,961.26 + 1,024.65, our system says it is an invalid numbers. In otherwise formatted numbers losses it numeric type data status and converted in terms of char / varchar2 data type. Hence we need to convert this char data type to numeric form and then we have to add using the syntax : “select to_number('1,961.26','99,99,999.99') + to_number('1,024.65','99,99,999.99') as total from dual” will produce correct result as

TOTAL

2985.91

TRANSFORMATION:

Function, which changes its object can be called a transformation. These functions are TRANSLATE, DECODE

We want to display descriptive report name against stored report_type column. In our example 'BS', 'PL' has been stored in report_type column refers to Balance_sheet and Profit and Loss Account. Let us display report_type and report description using the function decode (). The parameters are To be verified column_name, value of column, if that value is true then what to display, second value, if second value true what to display and else)

Decode(value, if1, then1, if2, then2, else)

In this case value → report_type

If1 → 'BS'

Then1 → 'Balance Sheet'

Else → 'Profit and Loss'

14.05 How to use decode() function

Syntax: select report_type,
decode(report_type,'BS','Balance Sheet','Profit and
Loss') as report_Desc from raak_ledger_master ;

Result:

REPORT_TYPE	REPORT_DESC
BS	Balance Sheet
PL	Profit and Loss

14.06 How to use case statement

The same function result can be achieved through case since it is simpler and understand and wordier commas are removed .

Case (column_name, when , then, when, then,
else end)

Syntax: select report_type, case report_type when 'BS'
then 'Balance Sheet' else 'Profit and Loss' end as
report_Desc from raak_ledger_master ;

REPORT_TYPE	REPORT_DESC
-------------	-------------

BS

Balance Sheet

PL

Profit and Loss

Before we like to see some of the conditions like grouping we shall know the existing records in raak_ledger_master

Syntax: select * from raak_ledger_master

Result:

RECORD_ID	LEDGER_CODE	LEDGER_NAME	REPORT_TYPE
-----	-----	-----	-----
1	BS-0001-00	CASH ACCOUNT	BS
2	PL-0002-00	SALES ACCOUNT	PL
3	BS-0004-00	CAPITAL ACCOUNT	BS

On seeing the above, we shall find that 'BS' found three times and 'PL' as 1 time. Now we like to have report_type and how many records are available

14.07 How to use group by clause

Syntax: select report_type, count(*) from raak_ledger_master group by report_type ;

Result :

REPORT_TYPE	COUNT(*)
-----	-----
BS	2
PL	1

We shall try to analyze, using the report_type column, as how many records are duplicated . Behind logic, count(*) producing more than 1 indicates that the records are duplicated.

14.08 How to use having clause

Syntax: select report_type, count(*) as records from raak_ledger_master group by report_type having count(*) > 1 ;

Result:	REPORT_TYPE	COUNT(*)
	-----	-----
	BS	2

14.09 How to use order by clause

On introduction of GROUP BY, HAVING command we shall be able to evaluate the records in different manner and meet different requirements. The resultant values

may not ordered one. In case if we require the values in ascending order or in descending order we shall user ORDER BY follows with column name.

Syntax: select * from raak_ledger_master order by ledger_name desc ;

Result:

RECORD_ID	LEDGER_CODE	LEDGER_NAME	REPORT_TYPE
-----	-----	-----	-----
2	PL-0002-00	SALES ACCOUNT	PL
1	BS-0001-00	CASH ACCOUNT	BS
3	BS-0004-00	CAPITAL ACCOUNT	BS

Desc (descending order need not be mentioned, if required ascending order).

15 – HOW TO CREATE SEQUENCE

One can assign unique numbers , such as record_id, to columns in any data table by using a sequence. For this purpose, one need not have separate table and code to keep track of unique numbers just another object like table, SEQUENCE object will serve our purpose. One could have seen the row number in excel / spreadsheet file can be equated with sequence.

As you are aware “create” command is used for creating objects, the same can be used for creating another object “sequence” as we did earlier for table.

```
“create    sequence    <<user_given_sequence_name>>
increment by 1 start with 1000”
```

In a nutshell A sequence is the database object used to generate UNIQUE INTEGERS for use as PRIMARY KEYS.

This sequence number will be used in the table where the column declared for its uniqueness. For example, we create RecordId sequence using the command as

“CREATE SEQUENCE "RECORD_ID" MINVALUE
1 MAXVALUE 99999999999999999999999999999999
INCREMENT BY 1 START WITH 1000 CACHE 20
NOORDER NOCYCLE “ If we use the command as
RecordId.Nextval will fetch the incremented value as
1001 and RecordId.Currval will fetch 1000. Once we use
NextVal command, sequence will internally incremented
and it will maintain uniqueness to supply next number.

Using Apex Object Browser, creation of sequence is as
follows:

Create SEQUENCE

Click CREATE in Object Browser → select the
Sequence Object

- ➔ Type Voucher_seq in Sequence Name
- ➔ Give minimum value as 1
- ➔ Give Maximum value as
99999999999999999999999999999999
- ➔ Give increment value as 1 CLICK NEXT
- ➔ Under Sequence window CLICK
CREATE

Every sequence number has been defined / generated by user. In between tables, usage of multiple sequence number may get overlapped and may lose the uniqueness in some of the table. For Example, in this application we have created four tables and all the tables has got unique column as record_id. On some occasions, cycle of sequence may produce may hardness. Hence, if one has decided to use unique sequence as record identifier instead of linking between tables, there is another functions called sys_guid() can be used in place of sequence. SYS_GUID() generates and returns a globally unique identifier made up to 16 BYTES but the data type is RAW and need to be converted in terms of number to use in our record_id.

The syntax of usage sys_guid() is as follows:

```
To_Number ( SYS_GUID( ),  
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX')
```

In this context, we have come across a new word as “Primary Key” which is nothing but identifying a record with a unique key value. This will be discussed at the time of creation of database constraint.

Now we shall see the tables we have created.

1. Table Name: raak_ledger_master

Column_name	Data Type / Width
Record_id	Number
Ledger_code	char(10)
Ledger_name	varchar2(200)
Report_type	char(2)

2. Table Name : raak_trans_master

Column_name	Data Type / Width
Record_id	Number
voucher_type	char(1)
voucher_number	char(10)
voucher_date	date,
voucher_naration	varchar2(400)
debit_total	number(16,2)
credit_total	number(16,2))

3. Table Name : raak_trans_detail

Column_name	Data Type / Width
-------------	-------------------

Record_id	Number
voucher_number	char (10)
ledger_code	char(10)
voucher_naration	varchar2(400),
debit	number(16,2),
credit	number(16,2))

In these three tables, you can find one common named column as record_id. This column is not going to be used on linking with other table. This column need to be populated by the system and not by the user, further no duplication is permitted and its unique nature needs to be maintained. Hence, Record_Id_seq will be useful on populating these tables. No harm in using database object sequence named record_id_seq in multiple tables. Further Ledger_code, Voucher_number columns in varied tables will also require sequence number but will be used with concatenation like VOU-0001, BS-0002-00 in CHAR column. At present, we require three sequences as record_id_Seq, voucher_seq, Ledger_seq. We will make an attempt to create these three sequences

Through object Browser	Through sql command
------------------------	---------------------

	Window
→ Sequences CREATE	Create sequence
→ sequences	“RECORD_ID_SEQ”
→ CREATE Sequence	minvalue 1 maxvalue
Window	99999999999999999999
→ record_id_seq	999999999
against	Increment by 1 start with
Sequence	1 nocache
name	Noorder nocycle
→ 1 against start with	
→ 1 against Minimum	
Value	
→ 100000000 against	
Maximum Value	
→ 1 against increment	
by	
leave other columns as	
Default → Next	
CREATE	

16 – HOW TO CREATE TABLE CONSTRAINT

Integrity constraint—a rule that restricts the values in a database.

- A **NOT NULL constraint** prohibits a database value from being null.
- A **unique constraint** prohibits multiple rows from having the same value in the same column or combination of columns but allows some values to be null.
- A **primary key constraint** combines a NOT NULL constraint and a unique constraint in a single declaration. That is, it prohibits multiple rows from having the same value in the same column or combination of columns and prohibits values from being null.
- A **foreign key constraint** requires values in one table to match values in another table.

- A **check constraint** requires a value in the database to comply with a specified condition.

16.01 **How to create Primary Key Constraint**

In every table a row / record need to be identified by one unique column. That column value can be defined and Primary Key. Creating a table without primary key is not advisable since any record need to be identified by some value. The primary key of a relational table uniquely identifies each record in the table. It can either be a normal attribute that is guaranteed to be unique or it can be generated by the DBMS (such as a globally unique identifier, or GUID). Primary keys may consist of a single attribute or multiple attributes in combination. Generally, the column declared as primary key will get populated from system generated sequence number. The various constraints will be proposed in any database, wherein the primary key is one of the constraints.

→ Object Browser select Tables caption under which select

→ `RAAK_LEDGER_MASTER`

- Constraints
- Create
- Change the default constraint name as Raak_LEDGER_Master_Pk as constraint name
- select Primary Key as Constraint Type
- select Record_id in Primary Column 1
- NEXT → FINISH

Script for creating Primary Keys:

Generally these types of the constraints will be created at the time of creating the table itself. If not done, then the table needs to be altered as follows:

```
“ALTER TABLE RAAK_LEDGER_MASTER ADD
CONSTRAINT "RAAK_LEDGER_MASTER_PK" PRIMARY
KEY ("RECORD_ID)”
```

For other two tables Table name, constraint name alone need to be changed since in both the tables, record_id is the primary key column.

Go to the Home → SQL Workshop → SQL Command Window ... type

```
ALTER TABLE RAAK_TRANS_MASTER ADD CONSTRAINT
"RAAK_TRANS_MASTER_PK" PRIMARY KEY
("RECORD_ID") CLICK RUN will alter the table
```

Go to the Home → SQL Workshop → SQL Command Window ... type

```
ALTER TABLE RAAK_TRANS_DETAIL ADD CONSTRAINT  
"RAAK_TRANS_DETAIL_PK"          PRIMARY          KEY  
("RECORD_ID") CLICK RUN will alter the table
```

Once we declare or define the primary key , it needs to be populated with sequence number generated by Oracle Sequence object through “TRIGGER”

16.02 HOW TO CREATE CHECK CONSTRAINT

It is another constraint altering the table structure. Hence, having created table one need to use ALTER TABLE command.

Required Business Rule for Ledge_Master: In Ledger_Master table, where in one column named as Report_Type and the data type declared as CHAR(2). Allowable data is only 'BS' or 'PL'. Which indicate BS as Balance Sheet and PL as Profit and Loss. If any other data tries to enter in the table database should reject the

action initiated. Hence we need to create the constraint as follows

- Object Browser under default table option
- select Raak_Ledger_Master
- Constraint → Create
- change the default constraint name as
RAAK_LEDGER_MASTER_CK1 against
constraint name
- select Check against constraint type
- select REPORT_TYPE against constraint on
column
- IN ('BS' , 'PL') in constraint expression
window
- NEXT → Finish

Equivalent SQL script

```
ALTER TABLE RAAK_LEDGER_MASTER ADD CONSTRAINT "RAAK_LEDGER_MASTER_CK1"  
CHECK ( "REPORT_TYPE" IN ( 'BS' , 'PL' ) )
```

Another constraint Ledger_Name should be left blank or NULL value is not permissible.

- Object Browser
- select Raak_Ledger_Master
- Constraint → Create
- change the default constraint name
 - as RAAK_LEDGER_MASTER_CK2 against
- constraint name
- select Check against constraint type
- select LEDGER_NAME against constraint on
 - column
- IS NOT NULL in constraint expression window
- NEXT → Finish

Equivalent SQL script

```
ALTER TABLE RAAK_LEDGER_MASTER ADD CONSTRAINT "RAAK_LEDGER_MASTER_CK2"
CHECK ( "LEDGER_NAME" IS NOT NULL )
```

To create ledger_code as uniqueness:

```
alter table raak_ledger_master add constraint
raak_ledger_master_uk1 unique (ledger_code);
```

To create ledger_name as uniqueness:

```
alter table raak_ledger_master add constraint
raak_ledger_master_uk2 unique (ledger_name);
```

We will try to create another constraint using sql script in Home → SQL Workshop → SQL Command Window the following script may be typed and executed so that voucher_type column will not accept other than 'J', 'P', 'R' which indicates Journal, Payment , Receipt.

```
ALTER TABLE RAAK_TRANS_MASTER ADD CONSTRAINT  
"RAAK_TRANS_MASTER_CK1" CHECK (  
"VOUCHER_TYPE" IN ( 'J' , 'P' , 'R' ))  
CLICK  
RUN will generate the constraint required.
```

16.03 HOW TO CREATE UNIQUE CONSTRAINT

In the raak_trans_Master, we have column voucher_number and it must be declared as unique value since it has got link with detail file. Before linking this key with detail file, this column must be created and checked as unique value. (No duplicate value is allowed).

- Object Browser under default table option
- select Raak_Trans_Master
- Constraint → Create
- change the default constraint name

RAAK_TRANS_MASTER_UK1 against
constraint name

→ select UNIQUE against constraint type

→ select VOUCHER_NUMBER against constraint
on column

→ NEXT → Finish

Equivalent SQL script

```
ALTER TABLE RAAK_TRANS_MASTER ADD CONSTRAINT  
"RAAK_LEDGER_MASTER_UK1" UNIQUE ( "VOUCHER_NUMBER" )
```

16.04 HOW TO CREATE FOREIGN KEY CONSTRAINT

A **foreign key** means that values in one table must also appear in another table. The referenced table is called the **parent table** while the table with the foreign key is called the **child table**. The foreign key in the child table will generally reference a [primary key](#) in the parent

table. A foreign key can be defined in either a CREATE TABLE statement or an ALTER TABLE statement.

There are two tables defined as raak_trans_master and raak_trans_detail. The linking key between these tables are voucher_number. In the raak_trans_master, only one record will be maintained for every voucher since voucher_number unique and multiple records will be maintained in raak_trans_Detail on the same voucher_number. In otherwise, if one wants to add record in raak_trans_file, the given voucher_number must exist in the raak_trans_master file otherwise transactions will be denied.

The foreign key constraint should be raised in detail file i.e. on raak_trans_detail.

- Object Browser under default table option
- RAAK_TRANS_DETAIL
- constraints
- create
- change the default constraint name as
 raak_trans_detail_fk1 as constraint name
- Foreign Key as constraint type

- select / click cascade delete
- select voucher_number as foreign key column
- Raak_trans_master reference table name
- Voucher_number as reference table column list
- NEXT FINISH

Equivalent sql script is as follows

```
“ALTER    TABLE          "RAAK_TRANS_DETAIL"    ADD
CONSTRAINT "RAAK_TRANS_DETAIL_FK1" FOREIGN KEY
("VOUCHER_NUMBER")                                REFERENCES
"RAAK_TRANS_MASTER" ("VOUCHER_NUMBER") ENABLE
```

17 – HOW TO CREATE TRIGGERS

Trigger means a device or small unit of snatch has been pulled to release fire arm is called Trigger. In Oracle, to initiate an action at some definite point, a code will be written which can be called as Trigger.

Triggers are commonly used to:

- prevent changes (wrong data entry in table)
- log / audit changes (user identity on last usage time)
- enforce business rules (e.g. less than 18 years aged person cannot give entry for date of joining since it is prohibited)
- execute business rules (e.g. notify a manager every time an employee's bank account number changes)
- replicate, enhance performance (e.g. on every purchase / sale the updating the stock figures on some other table and log maintenance)

The following are major features of database triggers and their effects:

- triggers can cancel a requested operation
- triggers can cause mutating table errors.

There are typically three triggering events that cause data triggers to 'fire':

- INSERT event (as a new record is being inserted into the database).
- UPDATE event (as a record is being changed).
- DELETE event (as a record is being deleted).

Structurally, triggers are either "row triggers" (Row triggers define an action for every row of a table or "statement triggers"(statement triggers occur only once per INSERT, UPDATE, or DELETE statement.)

Furthermore, there are "BEFORE triggers" and "AFTER triggers" which run in addition to any changes already being made to the database.

Triggers do not accept parameters, but they do receive information in the form of implicit variables. For row-level triggers, these are generally OLD and NEW variables, each of which have fields corresponding to the columns of the affected table or view; for statement-level

triggers, something like SQL Server's Inserted and Deleted tables may be provided so the trigger can see all the changes being made.

Trigger is a database object. This trigger can be initiated through APEX Wizards and as well as through sql scripts.

- Login Apex
- SQL Workshop
- Object Browser
- Select Triggers in the left Window
- CREATE
- select the table name RAAK_LEDGER_MASTER
- change the default constraint name as
RAAK_LEDGER_MASTER_BI as constraint name
- Firing Point as BEFORE
- Insert against "OPTIONS"
- Click Tick FOR EACH ROW
- Leave the when column option as blank
- Write the body as

```
:NEW."REPORT_TYPE" := UPPER(:NEW."REPORT_TYPE");  
:NEW."LEDGER_NAME" := UPPER(:NEW."LEDGER_NAME");  
IF :NEW."RECORD_ID" IS NULL THEN  
:NEW."RECORD_ID" :=  
    TO_NUMBER(SYS_GUID(), 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX');  
END IF;
```

- Click NEXT → FINISH

This trigger will generate system sequence number and will fill the record_id column. Any letters typed against report_type, ledger_name column will get converted as upper case. Record_id will have unique value even in transferring this table in to another database and it will not get

affected its uniqueness. Check constraint with respect to report_type will be checked after firing this trigger. On firing this trigger column will get uppercase letters. Further check constraint will check the validity of the data as 'BS' or 'PL' against report_type column.

The above wizard option could have achieved through the following script triggers for raak_ledger_master.

How to use before insert trigger

```
create or replace trigger "RAAK_LEDGER_MASTER_BI"
```

```
BEFORE
```

```
insert on "RAAK_LEDGER_MASTER"
```

```
for each row
```

```
begin
```

```
:NEW."REPORT_TYPE" := UPPER(:NEW."REPORT_TYPE");
```

```
:NEW."LEDGER_NAME" := UPPER(:NEW."LEDGER_NAME");
```

```
IF :NEW."RECORD_ID" IS NULL THEN
```

```
:NEW."RECORD_ID" :=
```

```
TO_NUMBER(SYS_GUID(),'XXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
XXXXXXXXXX');
```

```
END IF;
```

```
end;
```

We shall create the same system generated sequence script for raak_trans_master, raak_trans_detail also.

While writing the trigger body, you shall find two new words like NULL, :NEW. Null means no data. It is not equivalent to zero, or spaces. Hence comparison cannot be made with null data. But wherever, if you want to

compare null column with some other data, you shall give assumption value through one function command. For example If you want to write as If record_id = 1 then

In this case if record_id value is null then this comparison fails instead of that the following syntax may be used “if NVL(record_id, 0) = 1 then”. Here NVL function takes a role if the given parameter or column value is null or no data found then that may be replaced as zero.

Hence NVL() is function which will accept two parameters such as column name, field name, variable name, separated by comma and then the required default value may be given as second parameters.

:NEW. On seeing the word, we shall expect :OLD. Also. Generally :NEW. Followed by column name will indicate, new value of column name and :OLD.column name refers to old value available in the data storage.

For example, there is Pay as column name in a table and on particular record it has got 1200. On editing user has replaced this value with 1500.

:OLD.PAY → 1200

:NEW.PAY → 1500

another instance: at the time fresh record or on new insert, initial value may be null and then substituted value may be user's value

:OLD.PAY → null

:NEW.PAY → 1200

On this Before Insert Trigger with respect to Raak_Ledger_Master, we have one more requirement such as

→ Even if user enters ledger_name in lower case letters should be converted as upper case letters. Changes may be made on new record and

correcting the existing records also. Hence this trigger should get fired for converting the data into upper case during insert and update.

→ This conversion applicable for Report Type column too.

Very slight modification need to be done on Before Insert Trigger:

Before Modification of Trigger	After Modification
<pre> CREATE OR REPLACE TRIGGER "RAAK_ledger_MASTER_BI" BEFORE insert on "RAAK_ledger_MASTER" for each row begin if :NEW."RECORD_ID" IS NULL THEN :NEW."RECORD_ID" := TO_NUMBER(SYS_GUID() , 'X XXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXX') ; END IF; end;</pre>	<pre> create or replace trigger "RAAK_LEDGER_MASTER_BI" BEFORE insert on "RAAK_LEDGER_MASTER" for each row begin :NEW."REPORT_TYPE" := UPPER(:NEW."REPORT_TYPE"); :NEW."LEDGER_NAME" := UPPER(:NEW."LEDGER_NAME"); select :NEW."REPORT_TYPE" '- ' LPAD(LEDGER_SEQ.NEXTVA L,4,'0') '-00' into :NEW."LEDGER_CODE" from dual; IF :NEW."RECORD_ID" IS NULL THEN</pre>


```

:NEW."RECORD_ID" :=
TO_NUMBER(SYS_GUID(),'XX
XXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXX');
END IF;
end;

```

Please note the modifications: “insert on “ has been modified as “insert or update on” Two new lines has been added with respect to Ledger_name, Report_Type. Another new word / functions UPPER() has been used. This Upper() function will accept one parameter. For example Upper('raghu') will return value as RAGHU. Small / lower case letters has been embedded by single quotes in this function may be replaced by column name without single quotes.

Assume that there is only one record wherein 'bank account' has been entered in ledger_name.

If you give the command

```

select    ledger_name,    upper(ledger_name)    as    caps    from
raak_ledger_master;

```

will return

```

Ledger_name    caps

```

----- -----
bank account BANK ACCOUNT

Here, new word select is found. It is another command word like CREATE, ALTER SELECT means pick records. Select followed by column names and FROM is the must word to decide and should be followed with << table name >> .

Another business rule needs to be incorporated in the same trigger. At the time of inserting new record, Ledger_code should be generated and inserted in the Ledger code. Ledger Code length is 10.

Report Code			Ledger-Sequence				Group Code		
B	S	-	0	0	0	1	-	0	0

How to use before insert or update trigger

```
CREATE OR REPLACE TRIGGER "RAAK_ledger_MASTER_BI"
BEFORE
INSERT OR UPDATE on "RAAK_LEDGER_MASTER"
for each row
begin
if :NEW."RECORD_ID" IS NULL THEN
:NEW."RECORD_ID" :=
    TO_NUMBER(SYS_GUID(),'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX');
select :NEW."REPORT_TYPE" || '-' || LPAD(LEDGER_SEQ.NEXTVAL,4,'0') || '-00'
into :NEW."LEDGER_CODE" from dual;
END IF;
:NEW."LEDGER_NAME" :=
    UPPER(:NEW."LEDGER_NAME");
:NEW."REPORT_TYPE" :=
    UPPER(:NEW."REPORT_TYPE");
end;
```

In the above, you shall notice Ledger_code has been populated through the database sequence. Here .NEXTVAL increment the value after it has been used whereas if .currval followed with sequence name will produce the current sequence number and will not get incremented.

THE following trigger can be used for RAAK_TRANS_MASTER

CREATE OR REPLACE TRIGGER "RAAK_TRANS_MASTER_BI"

BEFORE

INSERT OR UPDATE on "RAAK_TRANS_MASTER"

for each row

begin

:NEW."VOUCHER_TYPE" := UPPER(:NEW."VOUCHER_TYPE");

:NEW."VOUCHER_NARATION" :=

UPPER(:NEW."VOUCEHR_NARATION");

if :NEW."RECORD_ID" IS NULL THEN

:NEW."RECORD_ID" := RECORD_ID_SEQ.NEXTVAL ;

select 'VOU-'||LPAD(VOUCHER_SEQ.NEXTVAL,6,'0') into

:NEW."VOUCHER_NUMBER" from dual;

END IF;

end;

The above trigger will help to convert upper case characters for of voucher_type, voucher_nARATION. Voucher_number will get populated using voucher_Seq created under database object. Dot and followed by nextval will increment the sequence number and padded with zeros with a prefix word as 'VOU-'. Here vou means voucher.

THE following trigger can be used for RAAK_TRANS_DETAIL

```

CREATE OR REPLACE TRIGGER "RAAK_TRANS_DETAIL_BI"
BEFORE
INSERT OR UPDATE on "RAAK_TRANS_DETAIL"
for each row
begin
if :NEW."RECORD_ID" IS NULL THEN
:NEW."RECORD_ID" := RECORD_ID_SEQ.NEXTVAL;
END IF;
end;

```

The following trigger will be created for balance_sheet table.

```

CREATE OR REPLACE TRIGGER "BI_BALANCE_SHEET"
before insert on "BALANCE_SHEET"
for each row
begin
if :NEW."RECORD_ID" is null then
select "RECORD_ID_SEQ".nextval into :NEW."REC_ID" from
dual;
end if;
end;

```

Trigger for Profit_loss table:

```

CREATE OR REPLACE TRIGGER  "BI_PROFIT_LOSS"
  before insert on "PROFIT_LOSS"
  for each row
begin
  if :NEW."RECORD_ID" is null then
    select "RECORD_ID_SEQ".nextval into :NEW."RECORD_ID"
from dual;
  end if;
end;

```

Having created three tables, various constraints, and triggers in different occasion, let us summarize on consolidated sql and lists :

```

CREATE TABLE  "RAAK_LEDGER_MASTER"
(
  "RECORD_ID" NUMBER,
  "LEDGER_CODE" CHAR(10),
  "LEDGER_NAME" VARCHAR2(200),
  "REPORT_TYPE" CHAR(2),
  CONSTRAINT "RAAK_LEDGER_MASTER_PK" PRIMARY KEY
("RECORD_ID") ENABLE,
  CONSTRAINT "RAAK_LEDGER_MASTER_CK1" CHECK (
"REPORT_TYPE" IN ('BS', 'PL')) ENABLE,
  CONSTRAINT "RAAK_LEDGER_MASTER_CK2" CHECK (
"LEDGER_NAME" IS NOT NULL) ENABLE
)
/

CREATE TABLE  "RAAK_TRANS_MASTER"

```

```

(      "RECORD_ID" NUMBER,
      "VOUCHER_TYPE" CHAR(1),
      "VOUCHER_NUMBER" CHAR(10),
      "VOUCHER_DATE" DATE,
      "VOUCHER_NARATION" VARCHAR2(400),
      "DEBIT_TOTAL" NUMBER(16,2),
      "CREDIT_TOTAL" NUMBER(16,2),
      CONSTRAINT "RAAK_TRANS_MASTER_PK" PRIMARY KEY
("RECORD_ID") ENABLE,
      CONSTRAINT "RAAK_TRANS_MASTER_UK1" UNIQUE
("VOUCHER_NUMBER") ENABLE
    )
/

CREATE OR REPLACE TRIGGER  "RAAK_TRANS_MASTER_BI"
BEFORE
INSERT OR UPDATE on "RAAK_TRANS_MASTER"
for each row
begin
if :NEW."RECORD_ID" IS NULL THEN
:NEW."RECORD_ID" := RECORD_ID_SEQ.NEXTVAL ;
:NEW."VOUCHER_NUMBER" := 'VOU-
'||LPAD(VOUCHER_SEQ.NEXTVAL,6,'0');
END IF;
end;
/

ALTER TRIGGER  "RAAK_TRANS_MASTER_BI" ENABLE
/

CREATE TABLE  "RAAK_TRANS_DETAIL"
(      "RECORD_ID" NUMBER,

```

```

        "VOUCHER_NUMBER" CHAR(10),
        "LEDGER_CODE" CHAR(10),
        "LEDGER_NAME" VARCHAR2(200),
        "VOUCHER_NARATION" VARCHAR2(400),
        "DEBIT" NUMBER(16,2),
        "CREDIT" NUMBER(16,2),
        CONSTRAINT "RAAK_TRANS_DETAIL_PK" PRIMARY KEY
("RECORD_ID") ENABLE
    )
/
ALTER TABLE "RAAK_TRANS_DETAIL" ADD CONSTRAINT
"RAAK_TRANS_DETAIL_FK1" FOREIGN KEY ("VOUCHER_NUMBER")
        REFERENCES "RAAK_TRANS_MASTER"
("VOUCHER_NUMBER") ENABLE
/

CREATE OR REPLACE TRIGGER "RAAK_TRANS_DETAIL_BI"
BEFORE
INSERT OR UPDATE on "RAAK_TRANS_DETAIL"
for each row
begin
if :NEW."RECORD_ID" IS NULL THEN
SELECT RECORD_ID_SEQ.NEXTVAL INTO :NEW."RECORD_ID" FROM
DUAL;
END IF;
end;

```

Totals of Raak_trans_detail (Voucher_Detail) Debit, Credit should get stored in raak_trans_master (Voucher Master) on each insert or update or delete on voucher details. Hence we shall create another after insert or update or delete trigger on raak_trans_details as follows insert or update trigger

How to use after insert or update trigger

```
create or replace trigger "RAAK_TRANS_DETAIL_AIUD"
AFTER
insert or update or delete on "RAAK_TRANS_DETAIL"
begin
update raak_trans_master set
debit_total = (select sum(nvl(debit,0)) from
raak_trans_detail
where raak_trans_detail.voucher_number =
raak_trans_master.voucher_number),
credit_total = (select sum(nvl(credit,0)) from
raak_trans_detail
where raak_trans_detail.voucher_number =
raak_trans_master.voucher_number);
end;
```

This concludes creation of these three table properties, leaving any business rules left on these tables and like to create new application with a play role of above tables. Before we proceed on creation of application, we will try to learn some more PL/SQL Concepts and creation of procedures, functions for some time.

18- HOW TO LAND IN PL/SQL

PL/SQL is Oracle's procedural language. Consolidated or combined sol's job may be classified as PL/SQL. Any business logic has been codified under this option. Generally, this has been group into blocks. Blocks may be named or anonymous. PL/SQL has got three sections such as "declarations", "executable commands", "exception handling". Variables are defined in the declarations section and starts with the word "declare", executable commands start with the word "begin" exception handling will start with "exception" and block should be terminated with "end".

We all familiar with declarations like

declare

```
l_temp0      number(16,2);  
l_temp1      varchar2(200);  
l_temp2      char(5);  
l_temp3      constant number(16,2) := 63.25;
```

CURSOR is another declaration type frequently used on PL/SQL. If anyone derives $5+3 = 8$ and here 8 is a result. It can be stored in variable called l_temp0. Whereas if you select all the columns of restricted rows or all the rows can be viewed as RESULT SET. Referring the result set by single name may be called as CURSOR. In otherwise result set of SQL may be termed as CURSOR DATA TYPE and should be named. Apart from this, declared data types can be assigned to memory variables declared under declaration section has got some roles like %TYPE,

%ROWTYPE. %TYPE inherits the definition of the column whereas
%ROWTYPE inherits all the column data types on a result set.

For example:

```
Cursor staff_cursor is select * from staff_master;
```

```
Staff_cursor_val      staff_cursor%ROWTYPE;
```

```
Staff_name_val        staff_cursor_val.staff_name%TYPE.
```

Here in the database staff_name has varchar2(200), hence staff_name_Val has been declared as varchar2(200). All the columns datatypes are assigned as in the table to this cursor staff_cursor_val. Another important role player command in PL/SQL is

```
If < some condition >
```

```
    then < some action>
```

```
elsif < some condition >
```

```
    then < some action>
```

```
else
```

```
    < some condition>
```

```
end if;
```

18.01 How to use if conditions

for example :

```
if avg_marks > 60 then
```

```
    result := "First Class" -- assigning some value to variable result
```

```
    elsif avg_marks > 50 then
```

```
        result := "Second Class" -- assigning values
```

```
    else
```

```
        result := "Not qualified" -- assigning values
```

```

        end if;
OR
If < some condition >
    Then
        If < some condition>
            then < some condition >
        end if;
    else
        < some condition>
    end if;

```

18.02 How to use nested if conditions

For example :

```

if level = 'Executive' then
    if pay >= 60000 then
        dear_allowance := 0.50    --- assignment of percentage
    end if;
else
    dear_allowance := 0.65    --- assignment of percentage
end if;

```

Usage of cursor and if conditions, we shall write a PL/SQL Block and see the results.

We have emp_id (employee Identification Number), basic_pay in table called emp_pay. In another table, emp_id, gross_pay and table named as emp_gross.

In the first table, pick records one by one and check whether basic_pay is less than 30000 or not. If basic_pay is less than 30000 then the gross_pay will be 175% of basic_pay else (more than 30000) gross_pay need to be calculated as 160%,

declare

low_pay constant number(3,2) = 1.75;

high_pay constant number(3,2) = 1.60;

l_gross_pay number(16,2);

cursor c1 is select * from emp_pay;

c1_val c1%ROWTYPE;

begin

 open c1;

 loop

 fetch c1 into c1_val;

 exit when c1%NOTFOUND;

 if c1_val.basic_pay <= 30000 then

 l_gross_pay := c1_val.basic_pay * low_pay;

 else

 l_gross_pay := c1_val.basic_pay * high_pay;

 end if;

 insert into emp_gross values (c1_val.emp_id, l_gross_pay)

 l_gross_pay := 0 ;

 end loop;

 close c1;

end;

18.03 HOW TO USE SIMPLE LOOP

```
loop
    exit when <<variable>> = 100;
end loop;
```

For Example:

```
declare
    some_val  number;
    cumulative_value  number;
begin
    some_val := 35 ;      -- initital value assigned
    loop
        exit when cumulative_value > 2000  -- sets the value to discontinue
    looping
        cumulative_value := cumulative_value + some_Val;
        some_Val := some_Val + 1;
    end loop;
end;
```

Here it will go on add as 35 + 36 + 37 + 38 and when the cumulative value exceeds 2000 this program quits and stop incrementing the some_val variable.

18.04 How to use cursor loop

CURSOR LOOP

```
loop
fetch  <<.....>> into <<.....>>;
exit when <<...>>%NOTFOUND;
end loop;
```

For example :

In one table (named as test1) has got two columns as rain_date and rain_fall.

We need to cumulate the rain_fall till the last record available. We are going to use one datatype as cursor along with looping concepts.

Declare

```
cum_railfall      number;
cursor C1 select * from test1;
C1_val C1%ROWTYPE;
```

-- C1 the name assigned to cursor

-- * symbol is used for selecting all the columns available in the test1 table.

-- C1_val is another variable takes the value of cursor C1 Record as

Rowtype

Begin

open c1; ----- opening the cursor

loop ---- repetitive action starts

```

    fetch c1 into c1_val;    -- every fetch will pick 1 record and assigns to
c1_val
    exit when c1%NOTFOUND;  -- when reads come to last record ... it
stops
    cum_rainfall := cum_rainfall + c1_val.rainfall;
end loop;
close c1;
end;
```

18.05 How to use for loop

```

for <<l_temp>> 1 .. 10 loop
end loop;
```

For Example:

We have one table named as test and we have two columns as col1 and col2. I want to insert 10 records. Col1 will have some values and col2 must be stored as square of col1. First value starts from 122.

```

declare
temp_Value number;
begin
for temp_value in 122 .. 131 loop
insert into test values ( temp_value, temp_value * temp_value);
end loop;
end;
```

18.06 How to use while loop

```

while <<l_temp>>  <= 99
```



```
loop  
end loop;
```

For example:

The previous program can be used as a while as follows

```
declare  
temp_Value number;  
begin  
temp_value := 122 ;  
while temp_value <= 131 loop  
insert into test values ( temp_value, temp_value * temp_value);  
temp_value := temp_value + 1 ;  
end loop;  
end;
```

19-HOW TO CREATE FUNCTION

Functions are forming part of set commands. Functions can return a value to the caller. Functions can be referred directly in the SQL queries. The value is returned through keyword within the function. Let us analyze the function syntax

```
create or replace function << function_name >>
( <<out_variable_name>> IN <<variable_data_type>> )
return <<data_type>>
<<variable_name>> <<data_type and length>>;
begin
    . . . . .
    . . . . .
    RETURN (<<out_variable_name>>);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20100,
            'some error messages');
end;
```

Now, we will write a complex function and learn some of the newer concepts of sql and concepts of function.

Purpose of the function : Any given number should be converted in terms of Indian rupee and paisa words. For Example Rs.123456789.12 should be converted into words as Rupee Twelve Crore Thirty-Four Lakh, Fifty-Six Thousand seven Hundred Eighty Nine and Paisa Twelve only. Normally JSP Format conversion of date will be done in million and billion whereas Indian rupee needs conversions Hundred Lakhs as One crore and no million terms. In fact One million need to be written as Ten Lakhs.

```
create or replace function "NUM_2_WORD"
(in_number in NUMBER)
return VARCHAR2
is
out_word varchar2(1000);
begin
select
'Rupees '||(case when nvl(crore,0)>0 then
to_char(to_date(crore,'J'),'JSP')||' Crore ' else ' ' end
)||' '||
(case when nvl(lakh,0)>0 then
to_char(to_date(lakh,'J'),'JSP')||' Lakhs ' else ' ' end
)||' '||
(case when nvl(units,0)>0 then
to_char(to_date(units,'J'),'JSP')||' ' else ' ' end )||'
'||
(case when nvl(paise,0)>0 then ' and Paise
'||to_char(to_date(paise,'J'),'JSP')||' ' else ' ' end
)||' '||

' Only ' into out_word
```

```

from
(select
trunc(rupee/10000000) as crore,
(trunc(rupee/100000) - (trunc(rupee/10000000) * 100)) as
lakh,
(rupee - (trunc(rupee/100000) * 100000)) as units,
paise as paise
  from
(select trunc(vall) as rupee, ((vall - trunc(vall))*100)
as paise from
(select in_number as vall from dual)));
return(out_word);
end;

```

Create or replace function needs a function name (user can assign reasonable name to suit to his logic interpretation and execution. Here We name our function name as Num_2_word since it converts numeric literals into word format. In the next line we need to give parameters within parenthesis with a word IN. (in_number IN Number). Function will return a value and nature of return value type need to be mentioned and in this case it has been mentioned as return varchar2. The word “return” is the must.

Return variable should be named with data length here in this case it has been mentioned as OUT_WORD varchar2(1000) and return statement also has varchar2.

20-HOW TO CREATE PROCEDURE

Oracle's Procedural Language (PL) is the superset of Structured Query Language (SQL). (For recap)

Structure of PL/SQL Block

Declare

<declaration section>

Begin

<executable commands>

Exception

<exception handling>

End;

Now, we will try to create small procedure will be switch position of given value. For example, if you submit the values as 10,20 after execution of logic will be displaying the given values as 20,10. While writing the procedure, the comments and explanations will be offered with the prefix of double dash '-' which is nothing but remark lines and will be ignored while execution of procedure commands.

```
-- create or replace is the command for creating or
replacing any
-- data base objects. In this case procedure is the
database
-- object and it need a name and here it is as
change_position.
```

```

create or replace procedure "CHANGE_POSITION"
-- Two temporary / field variables are declared as a1, a2
-- IN, OUT, IN OUT will be used for declaring whether to
--- receive input from system or to store or to update
followed
--- with data type. Note precision will not be given at
the
--- time of declaration.
(a1 IN NUMBER,
a2 IN NUMBER)
is
--- Any procedure should start with begin and must end
with end
--- notation.
begin
    declare
        first_number number;
        second_number number;
--- another procedure also can be created within the
declaration
--- section.
    procedure swaps ( num_one IN OUT NUMBER, num_two IN OUT
Number)
    is
        temp_num number;
    begin
        temp_num := num_one;
        num_one  := num_two;
        num_two  := temp_num;

```

```

end;

begin
    first_number := a1 ;
    second_number := a2 ;
--- dbms statement will be used to display the values
derived.
--- Always convert the date, number into char type before
--- display.
    dbms_output.put_line(first_number || ',' ||
second_number);
--- calling the procedure and while calling values are
passed
--- as parameters.
    swaps(first_number,second_number);
    dbms_output.put_line(first_number || ',' ||
second_number);
end;
end;

```

Using while loop and end loop procedure has been created to reverse the string given at the beginning.

**NEED: I want to reverse the given string such as
“RAGHUNATHAN” as “nahtanuhgar” using PL/SQL Block**

Declare

```

given_string varchar2(50) := 'RAGHUNATHAN' ;
repeat_loop_count number;

```

```

    result varchar2(50);
begin
--- length function is called for deriving the number of characters
stored
--- in the variable (given_string_varchar2(50))

    repeat_loop_count := length(given_string);
--- while command will decide whether to execute the following
commands
--- or not
    while repeat_loop_count > 0
--- loop and end loop is the syntax for doing exercises in repetetively
--- termination of loop will be decided by while command.

    loop
    result := result || substr(given_string,repeat_loop_count,1);
    repeat_loop_count := repeat_loop_count - 1 ;
    end loop;
    dbms_output.put_line(result);
end

```

RESULT:

NAHTANUHGAR

Statement processed.

0.00 seconds

In the ledger_master , number of rows is stored whereas 'BS', 'PL' two sets of values alone repetitively stored in Report type column. We like to select the unique records and do the previous exercise of reversing the content. In this case, pick out two values alone as 'BS', 'PL' from the entire table and reverse the contents as 'SB', 'LP'.

NEED: I want to select all the distinct unique records of account type from account_head_master and reverse the values and display

```
declare
    given_string varchar2(50);
    repeat_loop_count number;
    result varchar2(50);
    cursor cursor1 is select distinct account_type from
account_head_master
    order by account_type;
    cursor_val cursor1%ROWTYPE;
begin
    open cursor1;
    loop
```

```

fetch cursor1 into cursor_val;
exit when cursor1%NOTFOUND;
given_string := cursor_val.account_type;
repeat_loop_count := length(given_string);
while repeat_loop_count > 0
loop
result := result || substr(given_string,repeat_loop_count,1);
repeat_loop_count := repeat_loop_count - 1 ;
end loop;
dbms_output.put_line(result);
result := null;
end loop;
close cursor1;
end;

```

OUTPUT/RESULT :

LATIPAC

EUNEVER

Statement processed.

NEED: I want to know under what statement the account heads are accounted. For example capital account is accounted in balance sheet and sales account is accounted in profit and loss account etc. Use CASE statement to determine the results

declare

```

cursor cursor1 is select distinct account_head, account_type
from account_head_master
order by account_head;
cursor_val cursor1%ROWTYPE;
begin
open cursor1;
loop
fetch cursor1 into cursor_val;
exit when cursor1%NOTFOUND;
case
when cursor_val.account_type = 'CAPITAL' then
dbms_output.put_line(cursor_val.account_head || ' is accounted
in balance sheet');
when cursor_val.account_type = 'REVENUE' then
dbms_output.put_line(cursor_val.account_head || ' is accounted
in Profit and Loss Account');
end case;
end loop;
close cursor1;
end

```

OUTPUT/RESULT:

```

CASH ACCOUNT is accounted in balance sheet
DEPRECIATION is accounted in Profit and Loss Account
DISCOUNTS is accounted in Profit and Loss Account
EXPENSES is accounted in Profit and Loss Account
FIXED ASSET is accounted in balance sheet

```

INTEREST PAID is accounted in Profit and Loss Account

NEED: exercise of PL/SQL 003 may be achieved through decode function instead of case already illustrated.

declare

**cursor cursor1 is select distinct account_head, account_type from
account_head_master**

order by account_head;

cursor_val cursor1%ROWTYPE;

result varchar2(100);

begin

open cursor1;

loop

fetch cursor1 into cursor_val;

exit when cursor1%NOTFOUND;

**select decode(cursor_val.account_type,
'CAPITAL',cursor_val.account_head || ' is accounted in balance
sheet','REVENUE',**

**cursor_val.account_head || ' is accounted in Profit and Loss
Account') into result from dual;**

dbms_output.put_line(result);

end loop;

close cursor1;

end

OUTPUT/RESULT:

CASH ACCOUNT is accounted in balance sheet

DEPRECIATION is accounted in Profit and Loss Account

DISCOUNTS is accounted in Profit and Loss Account

EXPENSES is accounted in Profit and Loss Account

FIXED ASSET is accounted in balance sheet

INTEREST PAID is accounted in Profit and Loss Account

NEED: exercise of PL/SQL (Previous) may be achieved using if then else

declare

**cursor cursor1 is select distinct account_head, account_type from
account_head_master**

order by account_head;

cursor_val cursor1%ROWTYPE;

result varchar2(100);

begin

open cursor1;

loop

fetch cursor1 into cursor_val;

exit when cursor1%NOTFOUND;

if cursor_val.account_type = 'CAPITAL' then

result := cursor_val.account_head || ' is accounted in balance sheet';

elsif

cursor_val.account_type = 'REVENUE' then

result := cursor_val.account_head || ' is accounted in Profit and Loss

```

Account';
    else
    result := null;
    end if;
    dbms_output.put_line(result);
end loop;
    close cursor1;
end

```

```

OUTPUT/RESULT:
CASH ACCOUNT is accounted in balance sheet
DEPRECIATION is accounted in Profit and Loss Account
DISCOUNTS is accounted in Profit and Loss Account
EXPENSES is accounted in Profit and Loss Account
FIXED ASSET is accounted in balance sheet
INTEREST PAID is accounted in Profit and Loss Account

```

```

create or replace procedure "PL_BS"
is
begin
declare
pl number;
begin
delete from balance_sheet;
insert into balance_sheet (ledger_name) select ledger_name
from
raak_ledger_master where report_type='BS';
update balance_sheet set liability = (select

```

```

sum(nvl(credit,0)) from raak_trans_Detail
where raak_trans_detail.ledger_name =
balance_sheet.ledger_name);
update balance_sheet set asset = (select sum(nvl(debit,0))
from raak_trans_detail
where raak_trans_detail.ledger_name =
balance_sheet.ledger_name);
update balance_sheet set liability = (liability - asset),
asset= null where liability > nvl(asset,0);
update balance_sheet set asset = (asset - liability),
liability = null where asset > nvl(liability,0);
select sum(asset) - sum(liability) into pl from
balance_sheet;
if pl > 0 then
update balance_sheet set liability = pl where ledger_name
like 'ACCUMULATED PROFIT';
else
update balance_sheet set ASSET = pl*-1 where ledger_name
like 'ACCUMULATED LOSS';
end if;
delete from balance_sheet where nvl(liability,0) = 0 and
nvl(asset,0) = 0;
delete from profit_loss;
insert into profit_loss (ledger_name) select ledger_name
from raak_ledger_master
where report_type='PL';
update profit_loss set income = (select sum(nvl(credit,0))
from raak_trans_detail
where raak_trans_detail.ledger_name =
profit_loss.ledger_name);

```

```

update profit_loss set expenditure = (select
sum(nvl(debit,0)) from raak_trans_detail
where raak_trans_detail.ledger_name =
profit_loss.ledger_name);
update profit_loss set income = (income - expenditure),
expenditure= null where income > nvl(expenditure,0);
update profit_loss set expenditure = (expenditure -
income), income = null where expenditure > nvl(income,0);
pl:=0;
select sum(expenditure) - sum(income) into pl from
profit_loss;
--dbms_output.put_line(to_char(pl));
if pl < 0 then
update profit_loss set expenditure = pl*-1 where
ledger_name like 'PROFIT';
else
update profit_loss set income = pl where ledger_name like
'LOSS';
end if;
delete from profit_loss where nvl(income,0) = 0 and
nvl(expenditure,0) = 0;
end;
end;

```


21 – HOW TO MANIPULATE DATA INSERT / UPDATE / DELETE

Having created Ledger_Master, we shall try to create / insert records / rows into this table.

Let us understand the SQL script

```
INSERT INTO << Table Name >> ( <<column Names >> ) values ( << datas>> );
```

- column names may be given in multiple but should be separated with comma
- values need to be given in the same order separated by comma.
- character data type (Char, Varchar2) the values should be embedded by single quotes.
- In data's portion Function commands can be used like upper() etc.
- where condition can be given for restricting the insertion.
- the insertion can be made from another table
- the insertion can be from another table leaving the existing data's in the present table
- user can append records from another table.
- Column Name order need not be in the same order of creation

```
UPDATE <<TABLE NAME >> SET COLUMN NAME = <<VALUE >>  
WHERE <<CONDITION>>
```

- column_name and values can be separated by commas for multiple column update.

```
DELETE FROM <<TABLE NAME>> WHERE <<CONDITIONS>>
```

We shall see some of the examples for data manipulation

Let us create one small test table as follows with the following scripts

```
create table staff_master ( name varchar2(100), date_ob_birth date, pay
number(16,2));
```

How to insert / add / write records into a table / file

-- column values are in default order of creation

Now, we shall give the data in the same column order with which table created.

```
SYNTAX : INSERT INTO staff_master VALUES ( 'ABILASH' , '01-
JUL-1990', 40000.00)
```

RESULT: One row successfully inserted:

-- column values are in jumbled order

We like to give data in jumbled order like date_of_birth, pay and name

```
SYNTAX: insert into staff_master (date_of_birth, pay, name ) values ('17-
sep-1983', 37000, 'loreta chandy')
```

-- lesser number of column values given

We may like to give name and pay alone.

```
SYNTAX : INSERT INTO STAFF_MASTER (NAME,PAY )
VALUES ('KAVERY SINDIYA',17000)
```

-- column values fed, based on system prompt

Here we have to use bind variables. SQL_command window accepts : colon as bind variable operator and DOS SQL prompt will treat & ampersand symbol as bind variable.

```
SYNTAX:INSERT INTO STAFF_MASTER (NAME,PAY,DATE_OF_BIRTH )
VALUES (:NAME_PLEASE,:PAY_PLEASE,:DOB_PLEASE)
```

-- creation of another table along with existing data

SYNTAX:CREATE TABLE STAFF_TEMPST AS SELECT * FROM STAFF_MASTER

-- Adding records from another table

SYNTAX: INSERT /*+ APPEND */ INTO STAFF_MASTER (NAME, DATE_OF_BIRTH, PAY) SELECT NAME, DATE_OF_BIRTH, PAY FROM STAFF_TEMPST WHERE NAME NOT IN (SELECT NAME FROM STAFF_MASTER)

How to update records / rows into a table

-- updating one column on condition.

SYNTAX : UPDATE STAFF_MASTER SET PAY= 50000 WHERE PAY > 50000

-- updating column for all rows

SYNTAX UPDATE STAFF_MASTER SET PAY = PAY + 5000

How to delete records / rows in a table

-- deleting few rows on meeting condition

SYNTAX : DELETE STAFF_MASTER WHERE DATE_OF_BIRTH IS NULL

-- deleting all rows

SYNTAX TRUNCATE TABLE STAFF_MASTER;

22 – HOW TO CREATE Application

Login Oracle Apex

Select Application Builder

→ Click CREATE

→ Select CREATE APPLICATION

→ Click NEXT

→ default schema name may appear as application name and
change the Name of application as “RAAK 01”

→ Leave the default application number

→ select From Scratch option from Create Application

→ Leave the schema name as LEARNER click NEXT

Under ADD PAGE window, select the BLANK option against Select Page

Type Leave the Page Number as 1 Click ADDPAGE

Under Create Application window Click NEXT

→ select one level tabs and click NEXT

→ Click NO against copy shared components click NEXT

→ select application express under authentication scheme

→ Type DD-MM-YYYY under date format , leave all other default entries

→ click NEXT select Theme 1 under select theme click NEXT

→ confirm request by clicking CREATE

RUN the application Give username and password You will be landing in
the created First Page.

23 – HOW TO CREATE PAGE

Our Financial Accounting Application is web based applications and each screen will be developed as web pages and numbered. Hence Our initial design of Page Number and the Macro Contents are as follows:

Page Number	Action Plan
1	Main Menu – Master, Transaction, Report, Exit
2	Master Menu – Ledger Master, Inventory Master Return 2 Main Menu
3	Transaction Menu – Vouchers, Sales, Purchase Return 2 Main Menu
4	Report Menu – Financial Control Report, Inventory Status, Return 2 Main Menu
5	Tabular Form For Ledger Master which provides, insert, update, delete (This Page called from Page 2 and will return to Page 2)
6	Called from Page 3 Master Detail Form Master Form for Voucher_Master
7	Called from Page 7 Details of Voucher_detail in tabular form Will Return to Page 3
8	Called from 4 Generally Trail Balance, profit and Loss Account, Balance sheet will occupy different

	report pages. But, here we accommodate all the reports in this page in various regions. Return 2 Page 4
--	--

So far, using apex wizards, and using the default values, we were able to generate a financial accounting application package. Developed applications have got lots of bugs and loose ends on validation, accuracy etc. We need to tinker here and there to bring perfection. Of course, developed apex application was confidence booster. Let us explore some of the intricacies of apex application development which will facilitate and enrich our knowledge on development work. Here is the secret, that the developed application using wizard has generated internal coding which is nothing but PL/SQL itself. Now on, our Drill Down Exercises are begin

- Login Oracle Apex
 - select Application Builder
 - (while creating application itself, we have created default Page 1)
 - select RAAK (application)
 - Select 1 – Page 1

Now let us understand the Apex Page Properties. In page-1 properties has Oracle Application Express word and it can be replaced as company LOGO in our application.

Just Below there is a menu Bar which has Tabs (Menu Item) such as Home. Application Builder, SQL Workshop, Utilities. These tabs (menu items) are tagged or attached with horizontal bar which has a name as TABSET.

Below the tabs, on the left side, you shall notice
Home > Application Builder > Application 110 > Page 1 .

This will be names as Breadcrumb. In otherwise navigation bar (horizontally placed tree with branches. Home is a root) By clicking any one of the breadcrumb entry , you will be navigated to required area.

Entire Page has been divided into three major columns and has been named as PAGE RENDERING , PAGE PROCESSING , SHARED COMPONENTS.

In Each Column has got sections:

PAGE RENDERING	PAGE PROCESSING	SHARED COMPONENTS
Page	Computations	Tabs
Regions	Validations	List of Values
Buttons	Processes	Breadcrumbs
Items	Branches	Lists
Computations		Templates
Processes		Theme
		Security
		Navigation Bar

Page Rendering (Show Page)

The objects defined in the page are getting displayed at the time of Page submission. Generally, Pages accommodate regions and the style of regions is controlled by template. Various types of regions are inbuilt in the application development such as html, report, forms etc. etc. Buttons,

page items (fields) are sub element of regions and it can accommodate various types like text areas, radio groups, select lists, pickers etc.

Page Processing (Accept Page)

Various actions like click, enter, request will initiate the processes and associated activities may be termed as Page Processing. In otherwise executing the logic defined by developers on the instant of event occurs / happens. Major activates of any page processing involves validation, processes and branching. Execution of any processes and controlled or by order of sequencing. For example, when an item likely to get stored in a table and before that event it may require the validation to ascertain the data's impact or validation may even initiated before submission of page too or the validation may be required before branching to another page or to another controlled event.

Shared Components

Few of the tab (menu) items, breadcrumbs, Popup List of values may be commonly used between various pages. Hence Actions which serve for more than one pages may be classified under shared components. Generally, themes, templates are declared in shared components. Few of the readymade templates are available as

- Button Templates
- Calendar Templates
- label Templates
- List Templates
- Menu Templates

- Page Templates
- Region Templates
- Report Templates

On seeing the three columns, and just below the column name there were various icons are getting displayed. Each icon represents the section associated with that column. By clicking the Section icon, you will be navigated to that section where you may be required give attributes.

Various Page types are as follows:

Page Wizards are as Blank Type, Multiple Blank Pages, Reports, Charts, Form, Wizard, Calendar, Tree Login Page, Access Control, and Page Zero:

Now we shall try to create 8 blank pages and give name as under

Page -1	Main_Menu
Page -2	Master_Menu
Page -3	Transaction Menu
Page -4	Report Menu
Page -5	Ledger_Master
Page -6	Voucher_Master
Page -7	Voucher_Detail
Page -8	Control_Report

Since we have already created Page-1, the title of the Page-1 need to be modified as Main_Menu.

Click on Page-1

- Under Page Rendering Column, go to Page Section
- Click Page Title

→ You would have find the page title as 'Page 1' and change it as Main_Menu. Click APPLY_CHANGES

→ Page Name under Page section

→ Change it as Main_Menu. Click APPLIES CHANGES.

→ You will land in Home → Application Builder → Application nnn

Now you can find two pages are listed. One as Main_Menu and other 101 as login page. Now we require creating all the remaining 7 pages with the page name and title already provided.

→ Now you are in Home → Application Builder → Application nnn

→ Click Create Page

This will display various Page type icons under Create Page Window

→ Select Multiple Blank Pages click NEXT

It will display a tabular form under the headings of Create Pages.

Below that column Headings will be as Page, Tittle, Alias, Tab set. Pages may have some default numbers. Now change it as required and give titles, same title may be used in aliases and leaving blanks on tab_set.

Pages	Title	Aliases	Tab Set
2	Master_Menu	Master_Menu	
3	Transaction_Menu	Transaction_Menu	
4	Report_Menu	Report_Menu	
5	Ledger_Master	Ledger_Master	
6	Voucher_Master	Voucher_Master	
7	Voucher_Detail	Voucher_Detail	

8	Control_Report	Control_Report	
---	----------------	----------------	--

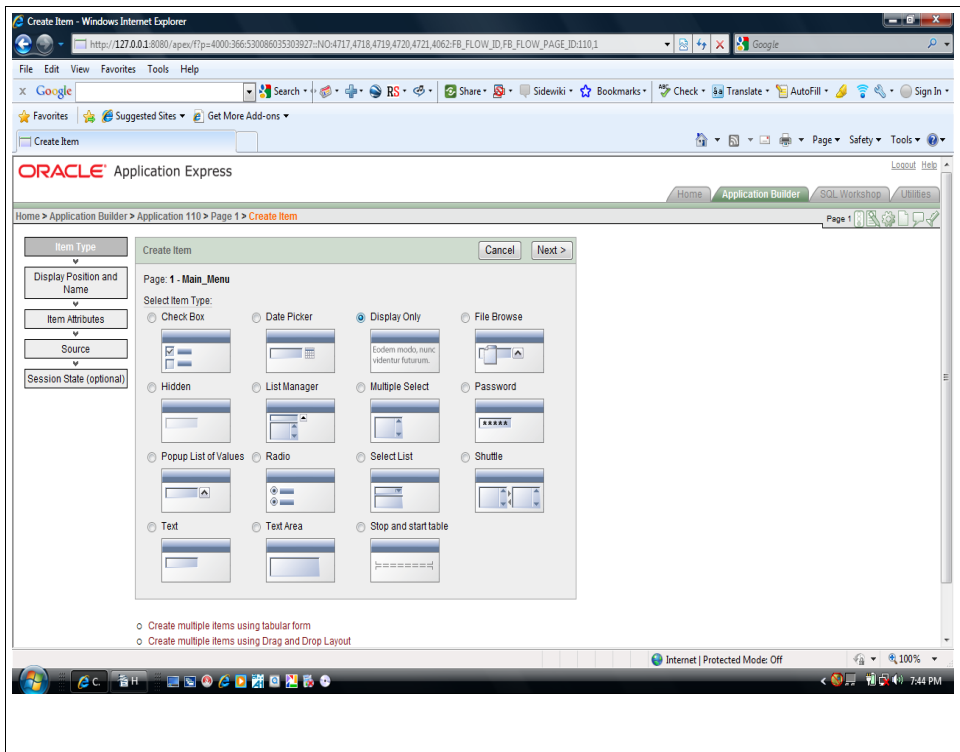
→ Click CREATE PAGES

Go to Home → Application Builder → Application nnn
you shall find all the eight pages created with aliases .

24-HOW TO CREATE PAGE ITEM

Login Oracle Apex using regular credentials of learner / learner / learner against workspace, scheme and password.

- Application Builder
- Application RAAK 01
- Click / Select Main_Menu
- select Item Section in Page Rendering Column
- Click Plus symbol for creation
- Create Item Window with the caption of Page 1 Main_Menu will get displayed. The following Icon Options will be displayed



- select display only Click NEXT
- leave the default on page type and click NEXT
- leave the default values against item name, sequence no.
- select Page 1 (1) 10 click NEXT
- Type Welcome 2 RAAK Application in the label and leave all

other

values as default click NEXT

- Leave the default in source window click NEXT
- leave the default values in session state and click CREATE ITEM

→ Run Application

→ Application login credentials like
 learner / learner against user and

password

and Login

Page 1 will get displayed .

As we have created on page item as P1_X, we shall the properties and see the result.

- Click Page 1 (Main Menu)
- Page Render column Under Items section click P1_X (item)

Now you are in Home > Application Builder > Application 110 > Page 1
> Edit Page item.

First section named as NAME, wherein there is a column Display as . This column may be filled using the select list. List will display various options

of Page Item such as check box, Date Picker, Text item, Text Area etc. Select Text Area with HTML Editor.

In the Label section, there is column Label wherein we shall give the value as “welcome to RAAK Application”

In the element section, there is a column named as HTML Form element attributes. Now we shall give the value as “style=background:RED;color:YELLOW;font-weight:BOLD;” Letters/digits will display in yellow color and background color will be red.

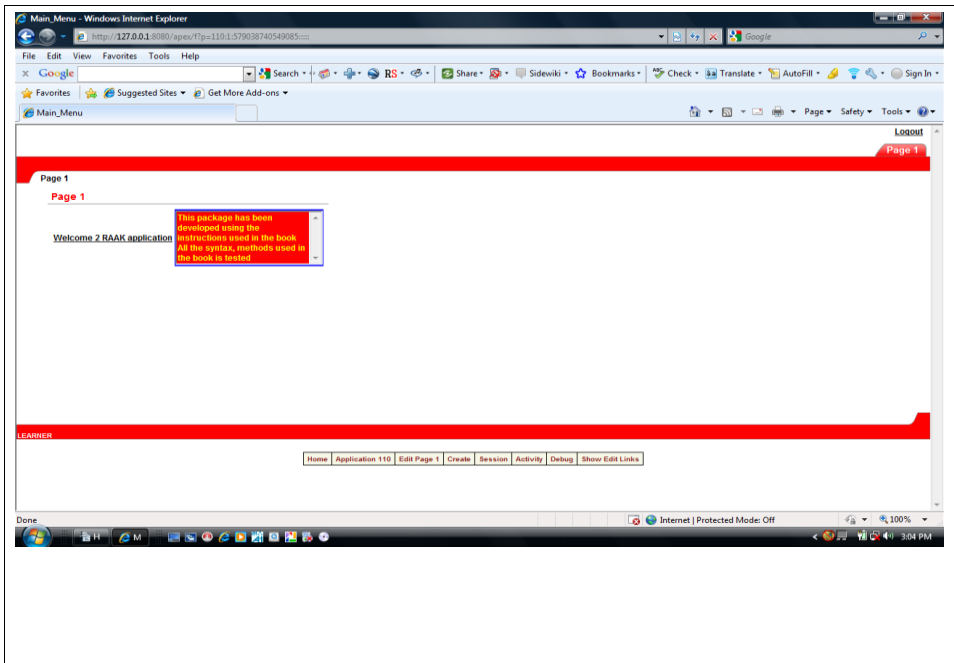
Below that you shall find Default section, where in the column named as default value. You shall type the some junk messages for testing purpose as follows:

“This package has been developed using the instructions used in the book All the syntax, methods used in the book I are tested”

Now click APPLYCHANGES and click RUN

You shall find the page1 with all the settings made in the above page 1.

Now, we understand the creation of page item, changing the attributes of the item.



In the resultant Page, you shall find Page1 displayed in three different places one at tab / menu item level, and another at region name level, and at breadcrumb level.

After learning the creation of region, tab , breadcrumb we shall try to assign reasonable name on all the above. Temporarily, we shall switch our activities to understand other aspects.

25-HOW TO CREATE LOGO

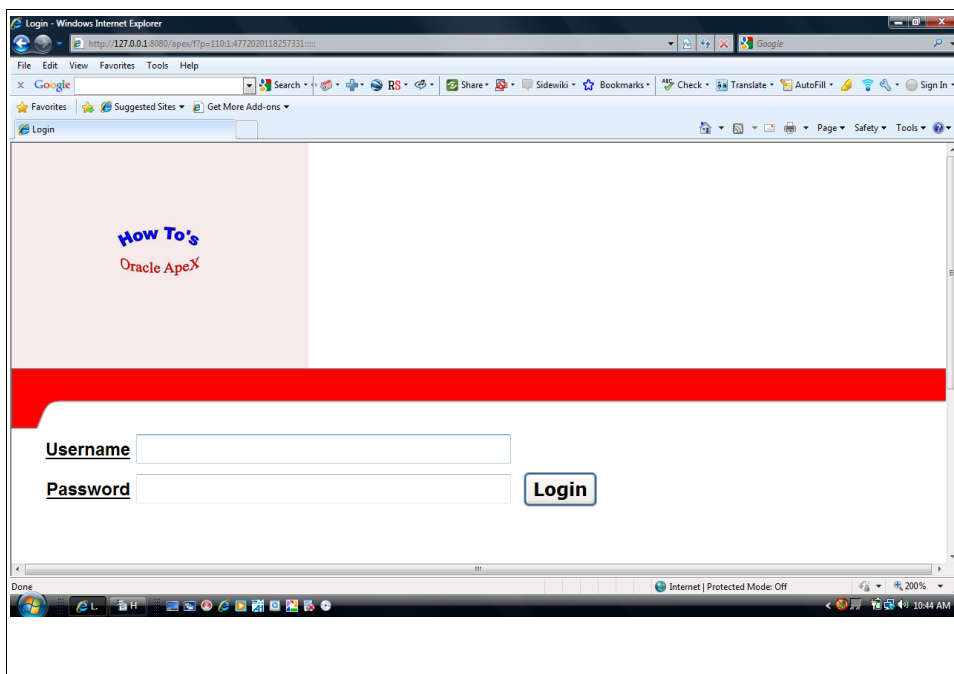
Any Image file like jpg, pif files can be made use in an application as LOGO of the applications, inventory products, or tab set images etc. Now we shall see how to accommodate LOGO in our application. For example I have one pif file stored in D drive in root folder as “raghu.pif”.

- Login Oracle Apex
- Application Builder
- Select Application
- select shared components
- Under various sections like Application, Logic, Security, Globalization, Navigation, User Interfaces, Reports, Files.....
select Images in Files option
- Under create Image browse and select the image file
- select from select list one of the options as
Workspace images/ Application images.
Workspace images will be made available for any applications whereas Application images will be made available only for particular application associated with selection
- Click Upload.

This will save the images in Oracle directory. Now select the stored images and link to the desired location of our pages. In this case, we like to set this image (RAAK_LOGO.JPG) as Logo in our application.

- select the application
- Shared Components
- select application definition
- Name, Availability, Global notifications, substitutions, Logo, Build options, Theme, Template defaults, component defaults will get selected
- select Logo section
- Type #APP_IMAGES#raghu.pif under image,
- width=75px; height=50px under Logo Attributes
- Click Apply Changes:

Now run the application, you shall find the logo sits above the tabset.



In the login page itself you shall find the logo How Tos JPG file and it will be appeared on all the pages we created since it has been created under application shared components. Our Next task is creation of tabs / Menu

26 HOW TO CREATE TAB / MENU

We need to create four tabsets and tabs within that tabset as Menu as follows.

In the Page 1 (Main_Menu)

→ Master_menu (On clicking this option control should be transferred to Page -2 (Master_menu)

→ Transaction Menu (On clicking this option control should be transferred to Page -3 (Transacton__menu)

→ Report- Menu (On clicking this option control should be transferred to Page -4 (Report_Menu)

For this we require Tabset (which is a holder for tabs /menu items) Tabset may be named as Main_Menu

Home

→ Application Builder

→ Application

→ Shared Components

→ Navigation

→ click Tabs

In the right side corner you shall find “Create New Standard Tab” and another option as Create New Standard Tab Set “ Menu Items can be called as tabs and the holder of tabs are called tabset. Now our task for page-1 Main_Menu, we shall assign tabset as Main_Menu and creation of tabs use the following data for Page 1

tabs name	Current tab for page
-----------	----------------------

Master_Menu	2
-------------	---

Transaction_Menu	3
Reports_Menu	4

Click ADD button (adjacent to Main_menu)

→ Type Master_Menu against Tab Label which is available in Create standard Tab Window Click NEXT

→Type 1 for Tab current for Page and click NEXT

→Leave the default sequence and Click NEXT

→Leave the default under Display conditions Window just click NEXT

→Under confirm window click CREATE TAB

Alternatively you can find two ADD buttons. Upper row ADD button is used for creation of new tab sets and bottom ADD button is used for creation of Tabs itself. When you are adding any new tabs, you shall find the select tabset on the upper row with the highlighter mark of green colour.

On Main_Menu Tab set, below that you click add button

type Master_Menu in tab label name

type 2 as tab current for page click next

leave all other things as default

click CREATE TAB.

In the Page 2 (Master_Menu)

Ledger_Master (On clicking this option control should be transferred to Page -5 (Ledger_Data Entry Form)

Inventory Master (On clicking this option control should be

transferred to Page -2 (Master_Menu)
since this menu tab item has been created for future development / up
gradation of our application software.

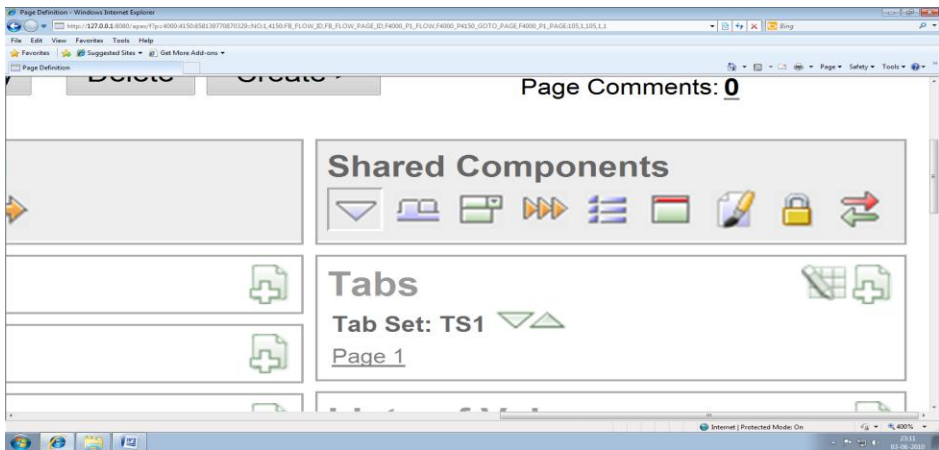
Return 2 Main (On clicking this option control should be
transferred to Page -1 (Main_Menu)

Tabset name for this menu/ tabs is Master_menu. This tabset along with
tabs will get displayed in Page-5 Ledger_Master data entry form too.

→ Edit Page 2

→ Shared Components → Tabs → CLICK Plus symbol for
creation

Click EDIT PAGE 2



First create New Tabset as Master_Menu wherein it will ask you to
define One tab and name it as Master_Menu and give current tab for page
no as 2. Then three more tabs as follows:

TabSet Name : Master_Menu

tabs name Current tab for page

Ledger_Master	5
Inventory_Master	2
Return_2_Main	1

The remaining two menus can be created using the following data's In the Page 3 (Transaction_Menu) Vouchers (On clicking this option control should be transferred to Page -6 (Voucher Data Entry) Purchase (On clicking this option control should be transferred to Page – 3 (Transaction_Menu) since this menu tab item has been created for future development / upgradation of our Application software.

Sales (On clicking this option control should be transferred to Page - 3 (Transaction_Menu) since this menu tab item has been created for future development / up gradation of our application software.

Return 2 Main (On clicking this option control should be transferred to Page -1 (Main_Menu)

Tabset name for this menu/ tabs is Transaction_menu. This tabset along with tabs will get displayed in Page-6 voucher_Master, Page-7 Voucher_detail data entry form too.

TabSet Name : Transaction_Menu

tabs name	Current tab for page
Voucher	6
Purchase	3
Sales	3
Return_2_Main	1

In the Page 4 (Report_Menu)

Control_Report (On clicking this option control should be transferred to Page -8 (Control Reports) Inventory Status (On clicking this option control should be transferred to Page -4 (Report_Menu)

since this menu tab item has been created for future development / up gradation of our application software.

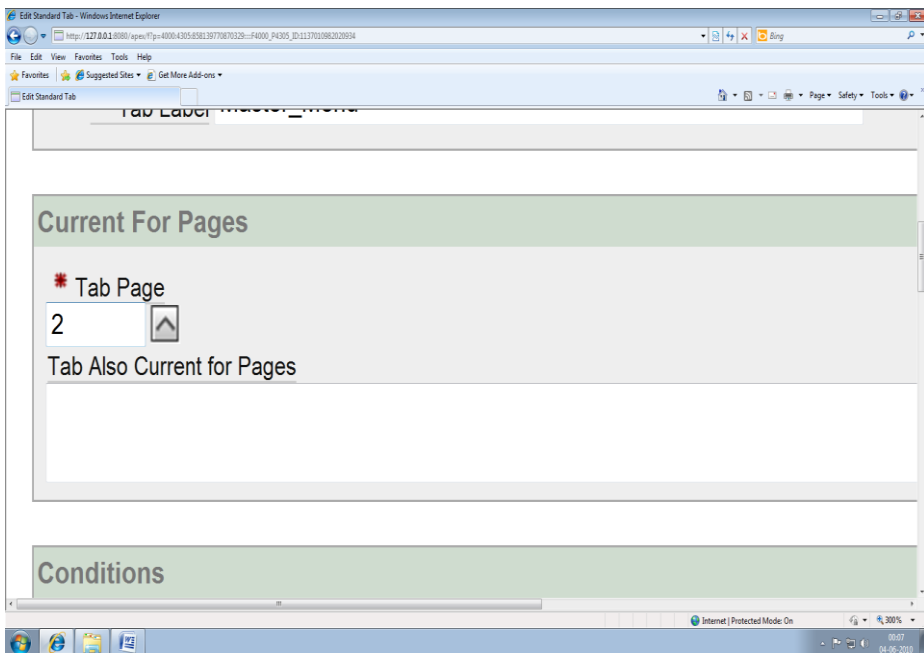
Return 2 Main (On clicking this option control should be transferred to Page -1 (Main_Menu)

Tabset name for this menu/ tabs is Report_menu. This tabset along with tabs will get displayed in Page-8 Control_report page too.

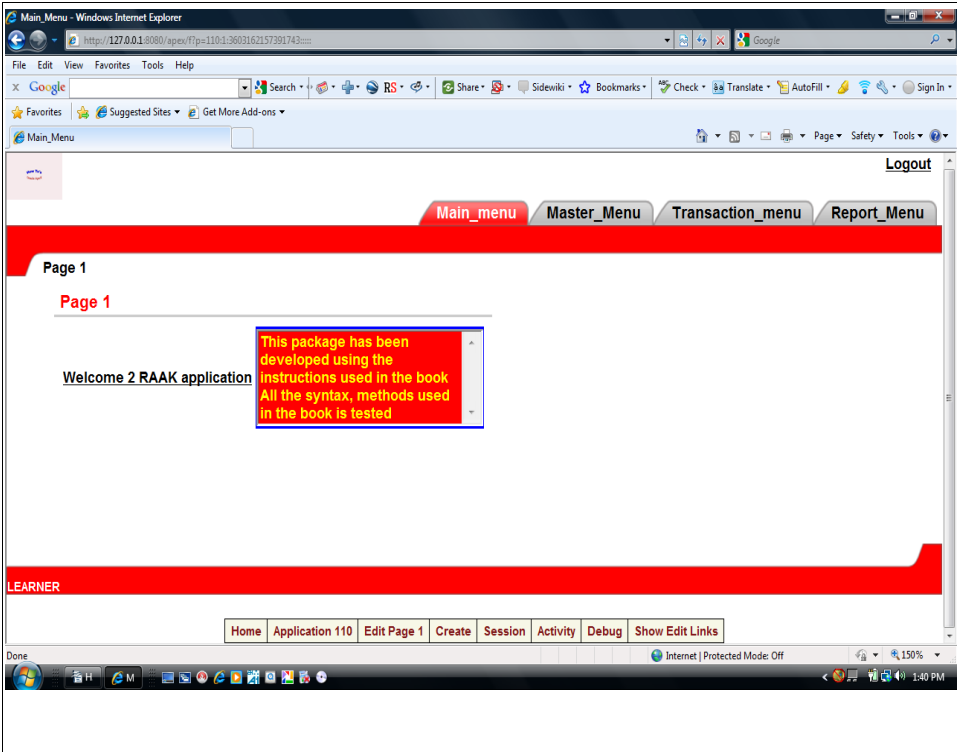
TabSet Name : Report_Menu

tabs name	Current tab for page
Control_Report	8
Inventory Status	4
Return_2_Main	1

Sometimes, after creation of tabset and tabs, required tabset may not get displayed in your required page. At that time edit required page use the following to set .. use the current set for pages option which will help us to get control.



Now you shall check by running the application, whether controls are getting shifted to required pages. For example In the Main Menu if you click Transaction Menu control will be shifted to Page 3 wherein if you click return 2 Main Menu tab, control will shifted back to page 1 (main menu). How is it?



27-HOW TO CREATE BREADCRUMP

Breadcrumbs are usually positioned horizontally across the top of a webpage below any title bars or headers linking back to each previous page through which the user navigated. Breadcrumbs provide a trail for the user to follow back to the entry point of a website. Generally, a greater than (>) glyph is used as hierarchy separator like

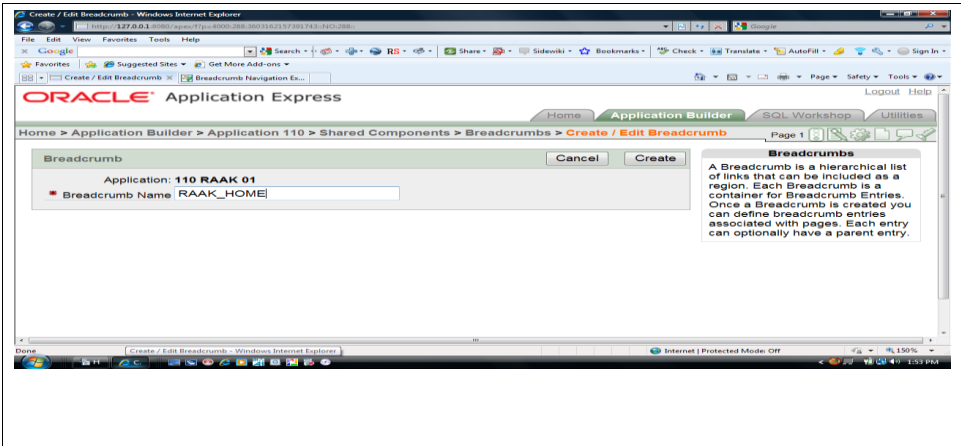
HOME > APPLICATION BUILDER > APPLICATION 110

Breadcrumbs provide hierarchical navigation to any number of levels. Once a breadcrumb is created, breadcrumb entries can be defined. To define breadcrumb entries click on the breadcrumb name.

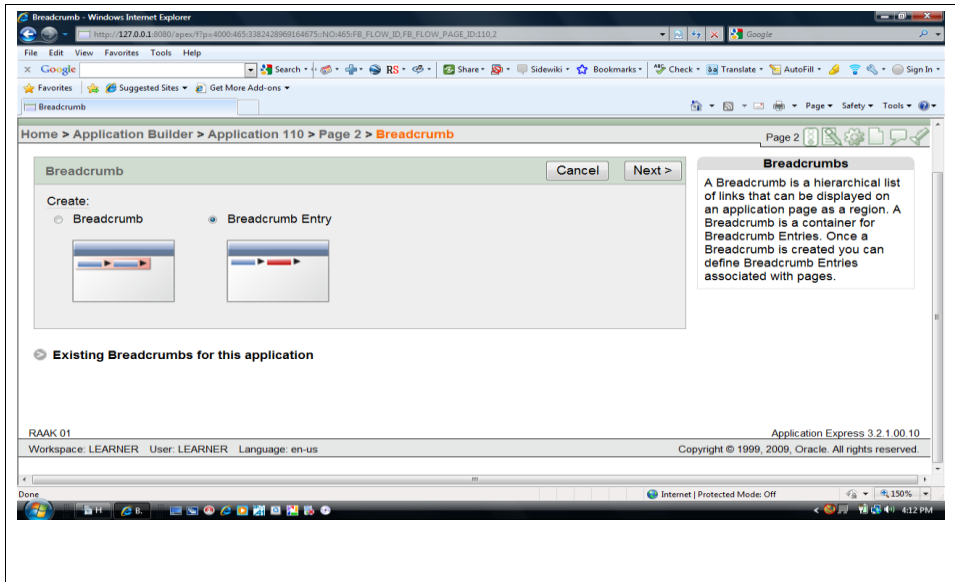
Breadcrumb entries are associated with pages and also identify a parent page. Breadcrumbs can be displayed on a page by adding a breadcrumb region using the create region wizard.

HOME > APPLICATION BUILDER > APPLICATION 110 >
SHARED COMPONENTS > BREADCRUMBS > CREATE / EDIT
BREADCRUMP

--



Now we will go to Page 2 (Master_Menu) create breadcrumb where in it will ask



Select breadcrumb Entry click NEXT

→ Leave the defaults on the Breadcrumb Container Region click NEXT

→ In the breadcrumb window, select the already created breadcrumb

RAAK_HOME, Hierarchical Menu against Breadcrumb Template

click NEXT

→ Under breadcrumb entry, by default it will select your page name and you have an option to change it, generally it is not advisable till you customize all the components, hence leave the default breadcrumb entry, but you select the parent entry, in this case HOME or (RAAK_HOME) will be the parent entry click NEXT

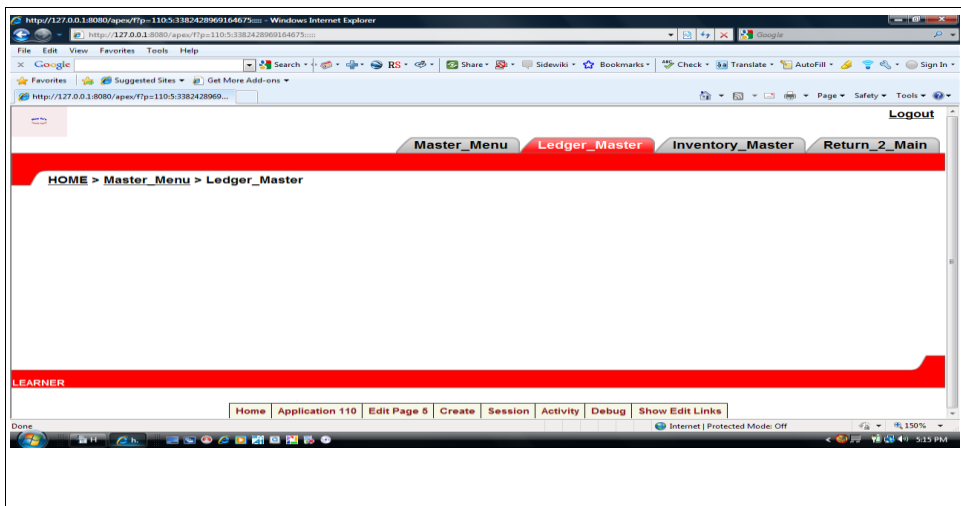
→ Click finish on Breadcrumb Confirmation Region

→ Now, you run the page to see the result. It is very unfortunate, your created breadcrumb entry found missing on your page. The reason is you have not created the breadcrumb region to accommodate the breadcrumb entries.

Every Page, you create breadcrumb region, and give associated linked page number and short form reference to be displayed. On Page Numbers and associated breadcrumb entries to be created as above are listed below:

Page Number	Breadcrumb Entry	Parent Breadcrumb
2	Master_Menu	Home
3	Transaction Menu	Home
4	Report Menu	Home
5	Ledger_Master	Master_Menu
6	Voucher Header	Transaction Menu
7	Voucher Detail	Voucher Header
8	Control Report	Report Menu

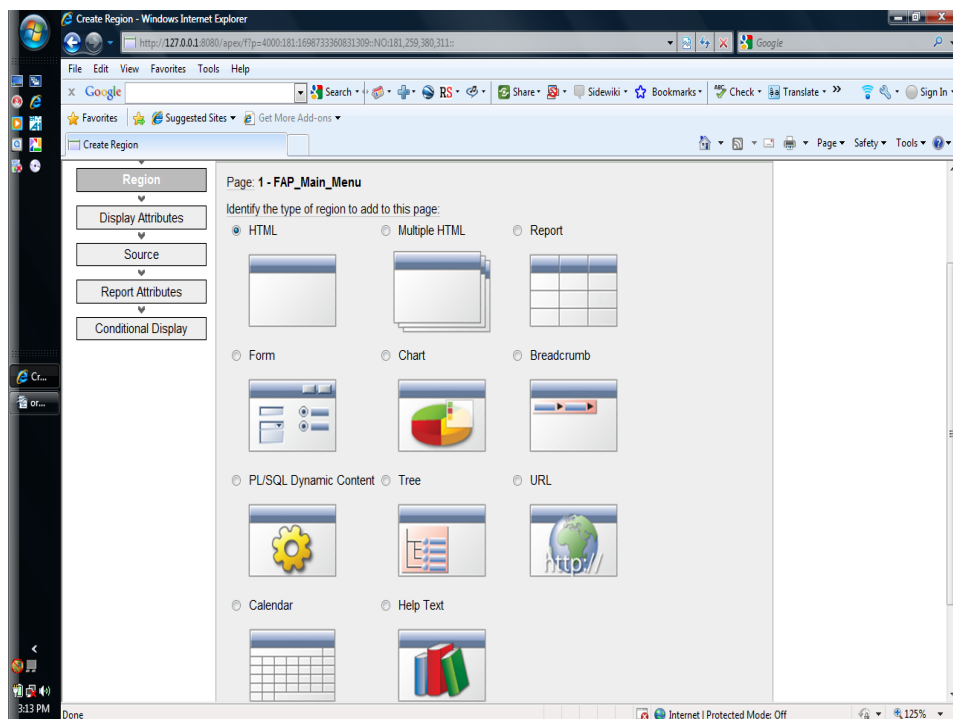
Having created the we shall see the breadcrumb display in page -5 as follows



RECAP : Any page creation of Breadcrumb will be easy by clicking Plus symbol under breadcrumb, it will ask Region to be created for breadcrumb, Breadcrumb entry and associated page to be displayed.

28-HOW TO CREATE REGIONS

As you are aware pages can accommodate many regions and few of the predefined region wizards are



To understand the regions, positioning of regions one must have conceptual view about themes and templates. The Main interface component of apex is nothing but themes. Any application can be associated with only one theme. Changing of theme at intermittent stage is not possible and at the time of creation it needs to be done. However parting theme from some other application to the proposed application is possible. Themes dictate templates, regions, and all other related

components for page rendering. Generally, template's function is to provide the user interface during the page render.

TEMPLATES AVAILABLE UNDER CURRENT REGIONS ARE REGIONS

borderless template:

no Template

Bracketed Region:

Breadcrump Region

Button Region with Title:

Button Region without Title:

CHART LIST REGION:

CHART REGION

FORM REGION:

HIDE AND SHOW REGION

LIST REGION WITH ICON

NAVIGATION REGION

NAVIGATION REGION ALTERNATE 1

REGION WITHOUT BUTTON AND TITLES

REPORT REGION WITHOUT TITLE

REPORT FILTER SINGLE ROW

REPORT LIST REGION

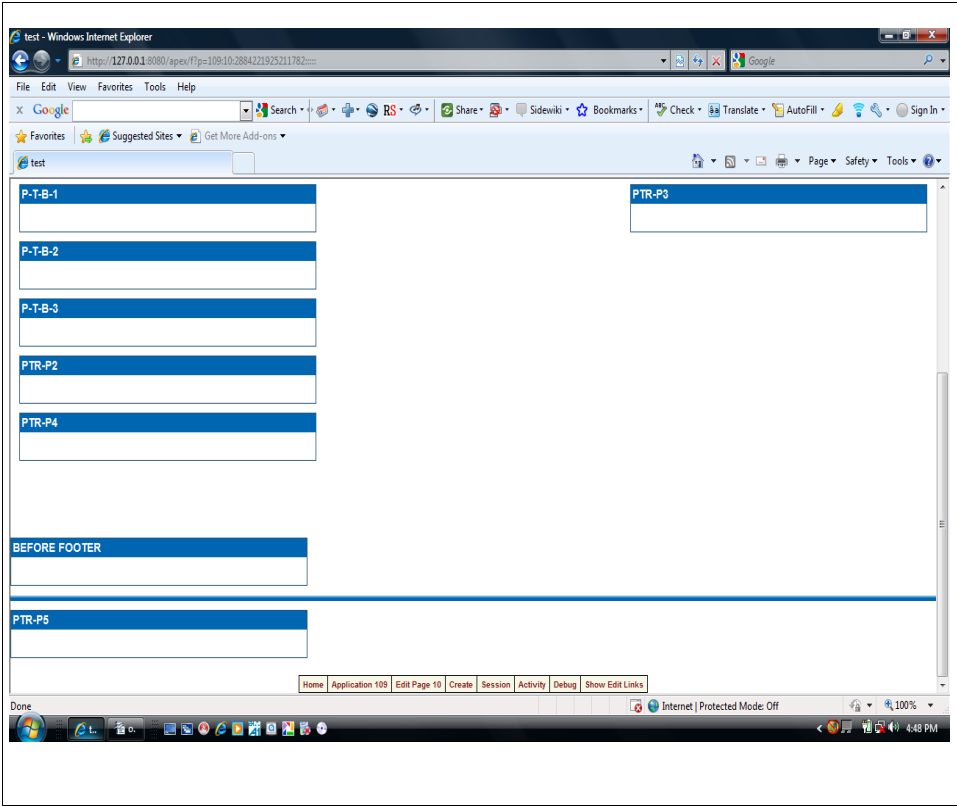
REPORT REGION

REPORT REGION WITH 100% WIDTH

SIDEBAR REGION

WIZARD REGION

POSITIONING OF REGIONS:



List of currently available display points on any web page

AFTER HEADER		
# LOGO		# NAVIGATION BAR #
TABSET AND TABS		
REGION POSITION 01		
	# GLOBAL_NOTIFICATION #	REGION

		POSITION 03
	# SUCCESS_MESSAGE #	
	#NOTIFICATION_MESSAGE#	
PAGE TEMPLATE BODY 01		
PAGE TEMPLATE BODY 02		
PAGE TEMPLATE BODY 03		
REGION POSITION 02		
REGION POSITION 04		
REGION POSITION 05		
REGION POSITION 06		
REGION POSITION 07		
REGION POSITION 08		

If you have selected two level tabs page template, Just below title, Region Position 06, Region Position 07, Region Position 08 will occupy before tabset and tabs.

In our application, our first task is creating Region, without which we cannot create, breadcrumb, button, page items etc. In Page 5 Ledger Master we are going to create tabular Form and while we create that page, we shall look into example on creating the regions since it is done with inbuilt wizards.

29 – HOW TO CREATE BUTTON

Immediately, we do not need any button still, we will create three button on page-1 (Main Menu) and on clicking that button we will be re-directed to Page 5 (Data entry form of Ledger Master), Page-6 (Voucher-Master data entry), Page-8 (Control Report Page).

- Home
- Application Builder
- Application 110
- Page 1
- Column Page Rendering
- Button Section
- Click Plus as create
- Under Create Button select the exiting region

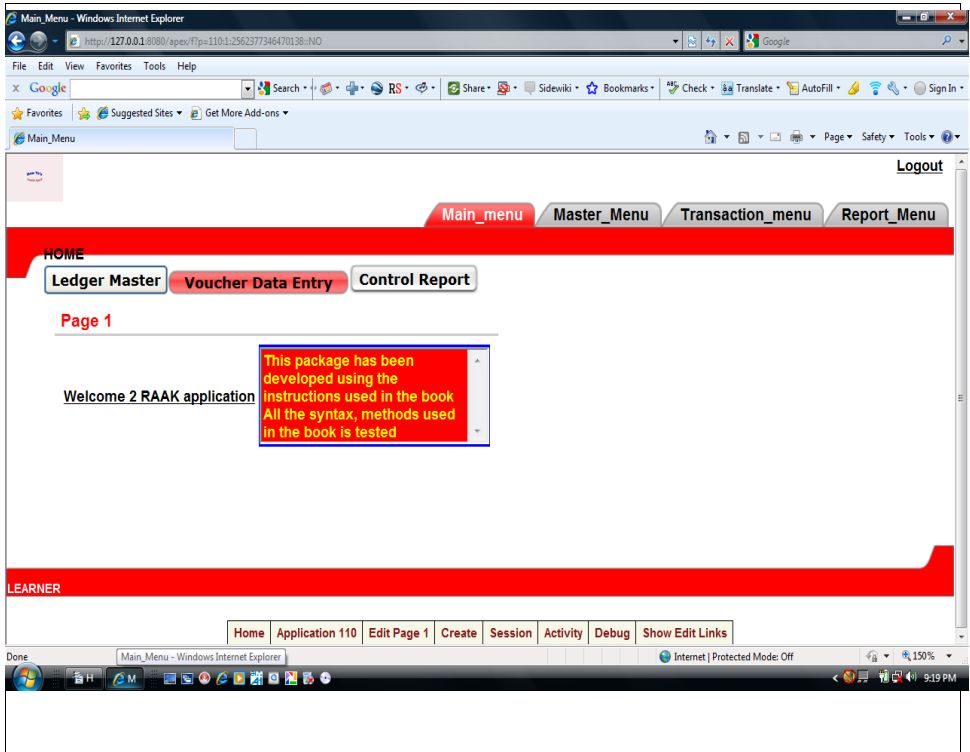
in this case we will select Breadcrumb region itself click NEXT

- Under Button Position Window, two options are getting displayed Create a button in a region position Create a button among the displayed region items we will select the option 1 and click NEXT

- Under Button attribute windows
- type Ledger_Master as Button Name
- type Ledger Master as Label

- select template driver against button type
- select “submit Page and redirect to URL” against
action option
- click NEXT
- Under image attributes window
- select Button Alternative 1 against Button Template
click NEXT
- Under Display Properties Window
- select button position as “Bottom Region”
- leave the default sequence
- leave the alignment as default
- select Ledger Master Page (5) against Branch to Page
click NEXT
- leave the default conditional display click CRETE BUTTON

Now we will repeat the same exercise to create two other buttons Voucher Entry and branch to page as 6 and Control_Report button with a branch to page 8. After creation of button on page 1, you shall the resultant display of button



You can note that Button Alternative 1 for Ledger Master, Button Alternative 2 for Voucher-Data-Entry, Button Alternative 3 as Control report resulted through button image attributes. On clicking, control will be shifted to respective pages.

30- HOW TO CREATE LOV

A List of Values can be referenced by page items as well as report fields. It controls the values displayed and limits the user's selection. Lists of Values can be static (based on values you enter) or dynamic (based on a SQL query)

While entering the data in the column, we get help through select list, list of values. There are two types of List values like creation from static values and creation from dynamic values. Now we will analyse two sets of List of values. There is a column Ledger_type in Ledger_master table. It can accept either 'BS, or 'PL' (two letter code). Whenever we go to that particular column, list of values should be vertically displayed as 'Balance Sheet Item' , 'Profit and Loss Account Item'. On selecting these two descriptive values, Codes should be returned and get stored in that column. This can be achieved through creation of LOV using static values.

While entering the voucher_details entry, system should provide Ledger Names vertically, and should return the value to that specified column. As and when we add entry on ledger_master, the dynamic values should get accommodated in the list of values. Hence creation of LOV using dynamic type is another choice.

- APEX HOME PAGE
 - APPLICATION BUILDER
 - APPLICATION 110
 - SHARED COMPONENTS
 - UNDER USER INTERFACE
 - LIST OF VALUES

→ create

Under LOV source Window

- select option as “From Scratch” against create list of values
click NEXT
- Under Name and Type Window
 - Type LOV_LEDGER_TYPE
 - select static type click NEXT

ORACLE Application Express

Home > Application Builder > Application 110 > Shared Components > Lists of Values > Create List of Values

Create Static List of Values

List of Values Name: **LOV_LEDGER_TYPE**

Sequence	Display Value	Return Value
1	BALANCE SHEET ITEM	BS
2	:OFIT AND LOSS ACCOUNT ITEM	PL
3		
4		
5		
6		
7		

Enter static values. Val order enter not display returned to engine. If y Return Va the Display

You can di attributes li controls an display by Values.

CLICK Create List Of Values

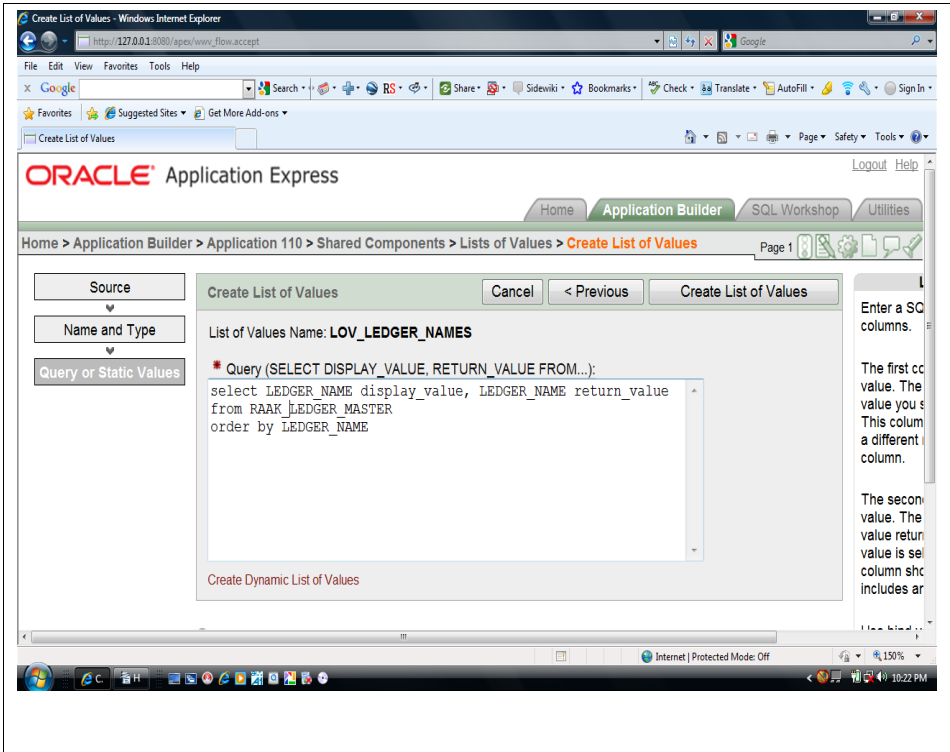
Now we will try to create another LOV using dynamic values concept

→ APEX HOME PAGE

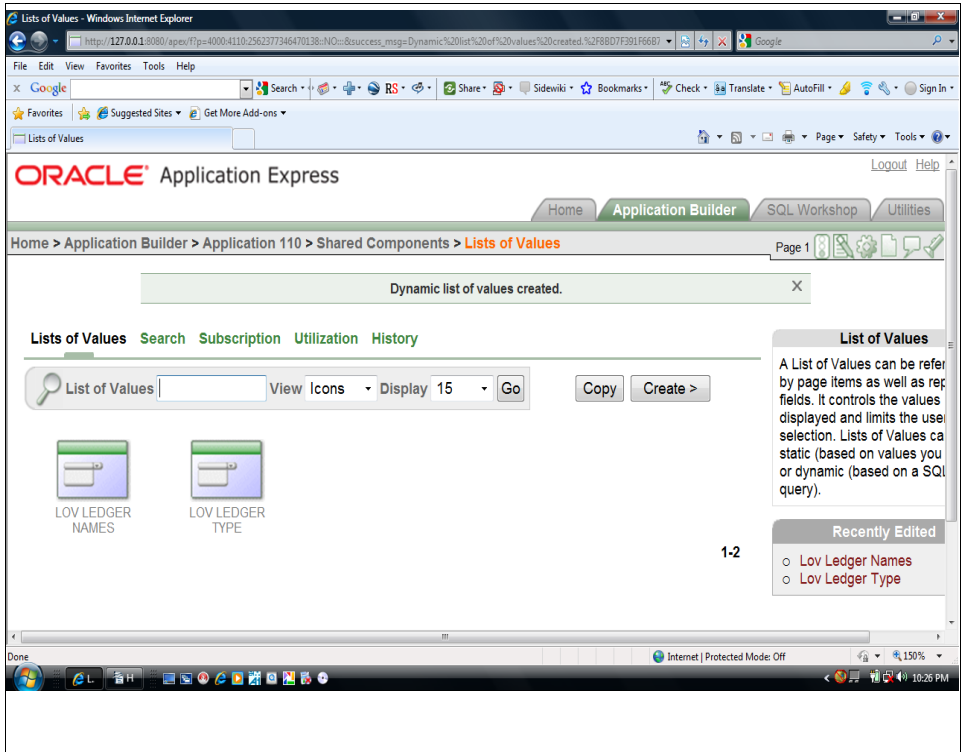
- APPLICATION BUILDER
 - APPLICATION 110
 - SHARED COMPONENTS
 - UNDER USER INTERFACE
 - LIST OF VALUES
 - create

Under LOV source Window

- select option as “From Scratch” against create list of values
click NEXT
- Under Name and Type Window
 - Type LOV_LEDGER_Name
 - select Dynamic type click NEXT



Click Create List of Values will result as follows



31-HOW TO CREATE COMPUTATION

A computation has been used to set / assign values to the page item during the page rendering or on page processing. The basic properties need to be set for the following.

- Computation Location
 - Item on the current Page
 - Item on another Page within the application
 - Application Level Item common to all Pages

(Global variables)

- Computation Point
 - Page Rendering Computation
 - Before Header
 - After Header
 - Before Region
 - After Region
 - Before Footer
 - After Footer
 - Page Processing Computation
 - After submit
 - Application Level Computations
 - On New Instance
- Computation Type

Let us see some of the computation example : Assume that telephone number need to be stored in a table has been captured in three pieces of elements like country code, area code, tele line number. Page

Item defined as :P1_COUNTRY_CODE, :P1_AREA_CODE, :P1_TELE_LINE. Data's are 91,4142,281700 needs to be stored in table as +91-4142-281700. Our process is nothing but concatenating these three piece elements into a single data element.

COMPUTATION TYPE → STATIC ASSIGNMENT

+&P1_COUNTRY_CODE. -&P1_AREA_CODE.-&TELE_LINE.

An ampersand symbol, page item name followed by period (.) will return the value stored and it has been joined with plus symbol and hyphen symbol to the required form.

COMPUTING TYPE → PL/SQL FUNCTION BODY

declare

l_temp varchar2(200) default null;

begin

l_temp := '+' || :P1_COUNTRY_CODE || '-' || :P1_AREA_CODE || '-'
|| :P1_TELE_LINE ;

RETURN l_temp;

end;

Bind variable colon (:) should be used while expression on concatenation

COMPUTATION TYPE → SQL QUERY

select '+' || :P1_COUNTRY_CODE || '-' || :P1_AREA_CODE || '-' ||
:P1_TELE_LINE from dual;

COMPUTATION TYPE → PL/SQL EXPRESSION

```
'+' || :P1_COUNTRY_CODE || '-' || :P1_AREA_CODE || '-' ||  
:P1_TELE_LINE
```

any valid expression used in sql statement or PL can alone be given in expression1 box for effecting computation.

→ Computation Conditions

→ condition is set for execution. For example if this page item is null then execute this computation etc. Normally, this will be enforced at the time of creating the computations.

32-HOW TO CREATE VALIDATIONS

→ Validation Method : SQL (compares page item with a data in table)

→ Select 1 from staff_mst where staff_code =

:P1_STAFF_CODE;

It checks whether the data exists in data.

→ Validation Method : PL/SQL (to validate complex logic)

→ begin

if :P1_STAFF_CATEGORY = 'ADMINISTRATOR'

AND :P1_STAFF_BASIC_PAY <=20000 THEN

RETURN FALSE;

ELSE

RETURN TRUE;

end if;

end;

normally this type of the validation comes under function returning boolean.

→ Validation Method : PL/SQL Expression

NOT (:P1_STAFF_CATEGORY = 'ADMINISTRATOR'

AND :P1_STAFF_BASIC_PAY <=20000);

→ Validation Method : ITEM NOT NULL

:P1_STAFF_CATEGORY (in expression 1 box)

Any column given in expression 1 box will be checked against not null status and pass the validations.

- Validation Method : ITEM string comparison
- select items in expression1 contained items in expression 2
- In expression 1 Box give the page item without bind variable
- In expression 2 Box give the data element separated by slash symbol for example ABC /DEF / GHU etc.

VALIDATION

- Column Page Processing → Section Validation
- Create Validation → Item Level Validation → NEXT
- Identify the item to be validated
- P1_STAFF_DOB (Date of Birth) NEXT
- Select a validation Method PL/SQL
- Select the type of PL/SQL validation you wish to create as PL/SQL Expression NEXT
- Leave the default with respect to sequence, validation name, error display location NEXT

In Validation Box

(sysdate - to_date(:p1_staff_dob,'dd-mm-yyyy')) >=18

In Error Message Box

18 YEARS SHOULD have Been COMPLETED BEFORE JOINING IN THIS FIRM click NEXT

Under condition Leave the default and Click CREATE

Another validation rule for P1_STAFF_DOJ

- At the time of joining one should have completed 18 years
- At the time of joining one should have not exceeded 58 years

→ Date of Joining should not be prior to 01-jan-2001 since the company formed / promoted on the said date

Upto validation Box, the earlier said procedure holds good except on the column name as P1_STAFF_DOJ instead of P1_STAFF_DOB

In Validation Box

```
(( ( to_Date(:p1_staff_doj,'dd-mm-yyyy') -  
to_Date(:p1_staff_dob,'dd-mm-yyyy') ) >= 18),  
to_date(:p1_staff_doj,'dd-mm-yyyy') >= to_Date('01-01-2001', 'dd-mm-  
yyyy'))
```

In Error Message Box

Either date of join falls before 01-jan-2001 or the date of join falls before completion of 18 years

Now we shall run the application and check whether the validation can get fired at the time of wrong entry itself.

33-HOW TO CREATE PROCESS

RECAP POINTS

- An application is nothing but set of Pages.
- A Page has been divided by three columns and contains the objects like Regions, buttons, tabs, items, etc.
- Three columns in a page are classified as Page Rendering, Page Processing, Shared Components
- Page rendering means is the process of generating a page from the database. Page rendering column has got all the controls and logics with which a page has been submitted or generated on rendered. In simple term SHOW PAGE is classified as PAGE RENDER
- Page Processing wherein all the computations, processes etc. has been evaluated and executed. In simple term ACCEPT PAGE is classified as PAGE
- Common components or objects shared by one or more pages are located in shared common components column.

Condition is the logic object which controls, display of regions, buttons, tabs, items, evaluating computations, validations and processes during page rendering and processing. Hence it is associated with all the objects listed in a page.

There are various types of conditions are available to implement in any one of the developed pages. But we shall try to see the popular and

frequently referred condition type with an example for understanding purposes.

→ Condition Type : Current Page in Expression 1

A user has given set of page numbers comma separated as 4,11,17. The given condition will be processed only when the rendered page numbers falls within the set of given numbers. In other words, page no 4, page no 11, page no 17 alone will evaluate this conditions and rest of the pages will ignore this condition while rendering or processing the pages.

→ CONDITION TYPE : EXISTS

Here, generally sql query will be given. For example “select 1 from staff_mst where staff_code = :P1_STAFF_CODE; This query must return a value otherwise this conditions will not get evaluated.

→ CONDITION TYPE : PL/SQL EXPRSSION

Any valid expression supported by PL/SQL can be given for evaluation. For example `NVL(:P1_STAFF_BASIC_PAY,0) > 10000`; This condition will be evaluated only when the value is more than 10000 and status marked as pass or this conditions determined as failed.

34- HOW TO CREATE BRANCH

- Branch is an instruction to go to a specific page or URL.
- Using the branch point we shall control the branches. The following points are available:
- On Submit : Before Computation Action will be performed as No computation, No validation, and No Process
- On Submit : Before Validation Computation part will be over but validation and process will not take effect before branching
- On Submit : Before Processing
- On Submit : After Processing
- On Load : Before Header

35-HOW TO CREATE TABULAR-FORM

Login oracle Apex using learner/learner

=> Application Builder

→ application

→ click Page 5

→ goto Page Rendering

→ goto Region section

→ click create button (Plus symbol)

→ select Form Region click Next

→ Click Tabular Form (create a data entry form

based on option) → Click NEXT

Under Identify table / view owner

→ allow the defaults

“learner” against table / view owner

“insert, update and delete” against allowed operations click NEXT

→ Select the RAAK_LEDGER_MASTER and click NEXT

→ Under Identify columns to display

→ select all columns and click NEXT

→ Leave the default Primary key as RECORD_id and click NEXT

→ select the existing trigger and click NEXT

→ select all the updatable columns and click Next

Under Page and Region attributes

Leave the Page No 5

Type LEDGER_MASTER against Page Name

Type LEDGER_MASTER_DATA_ENTRY_FORM against Region

Title

select Report Region against Region template

select Template standard against Report Template

select breadcrumb against breadcrumb

select Master_Menu under select Parent entry click NEXT

Under Tab (Optional)

allow the default button labels and click NEXT

allow the default branching page click NEXT

click FINISH under Tabular Form Confirmation

Linking this page No 5 to Page 2

Since Master Menu exists in Page No.2 and clicking ledger_MASTER tab control should be shifted to Page No.5 (Tabular form for Accounts Head Master created)

Edit Page No:2

→ Shared Components → Tabs → select Account_Head_master tab

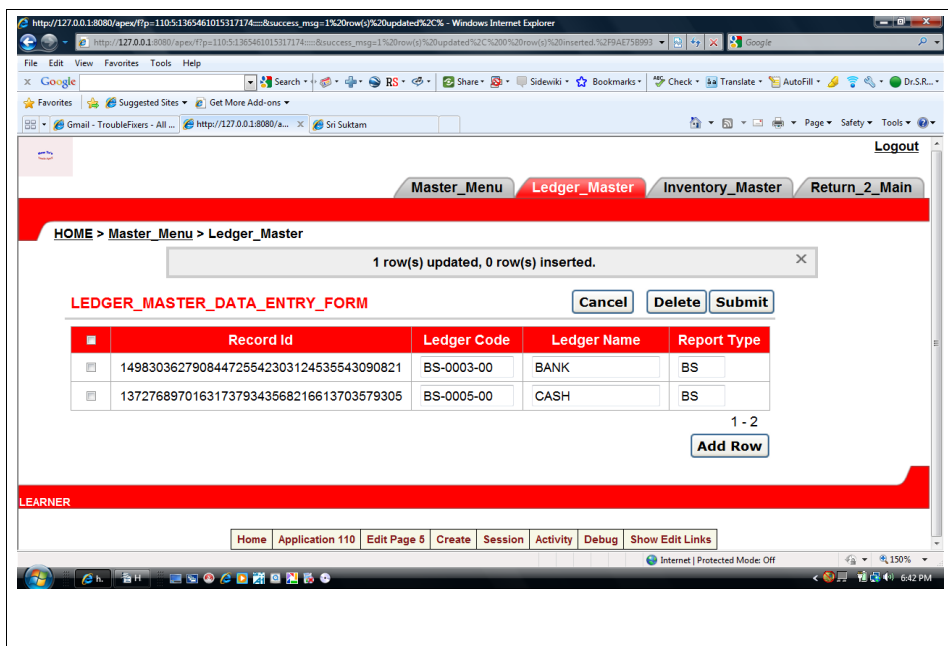
→ Under current for Pages select the Tab Page as 5 instead of existing page 2 (made at the time of creation temporarily) Click APPLY

CHANGES

Run the page 2 click the LEDGER_MASTER Tab will take you to Page 5.

Play with data and click Return 2 Page will take you to Main Menu.

At this stage you may need some sophistication on data selection like select list , popup, increasing width of data entry area, coloring, help status etc. etc.



While running the above page, we found that record_id need not be displayed since it is system generated number column and no manual intervention is required. Secondly Ledger_code is always generated from based on report_type, hence this column must be display only and not as enterable one. Report_type must have select list as Balance sheet item, profit and loss account item. Ledger_name column background color should be red and font color need to be as yellow: We will make an attempt to beautify the page tasks one by one as follows:

(A) Record_Id column should not display.

→ Edit Page 5

→ Under Page Rendering section region, select REPORT and click

→ Under Column Attributes

- Against RECORD_ID_DISPLAY aliases
- Untick the show radio button against show column
- click Apply Changes and Run the page (click railway signal)

(B) LEDGER_CODE column must be as display column (User should be able to access this column.

→ Edit Page 5

- Under Page Rendering section region, select REPORT and click
 - Under Column Attributes
 - Against LEDGER_CODE aliases
 - click edit attributes left side of the column icon with paper, pencil will be available.
 - select the tabular form element
 - display as text will get displayed change it using select list as “Display as TEXT (saves state)
 - click apply CHANGES and RUN
 - you will notice that column will not be available for users entry.

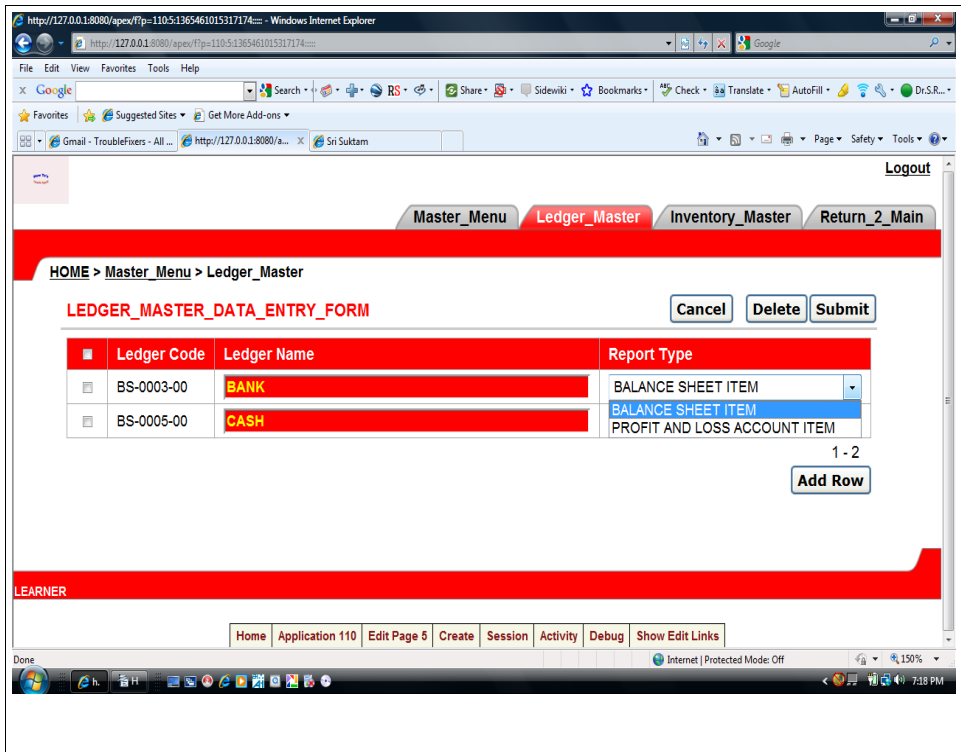
(C) LEDGER_NAME must have more width and background color as RED

Font Color as YELLOW.

→ Edit Page 5

- Under Page Rendering section region, select REPORT and click
 - Under Column Attributes
 - Against LEDGER_NAME aliases

- click edit attributes left side of the column icon with
paper, pencil will be available.
- select the tabular form element
- Element width will have default value as 16 and change it as 50
- Under element attributes
style = background: RED; color: YELLOW; font-weight: BOLD;
- click APPLYCHANGES and RUN.
- (D) REPORT_TYPE must have select list option to restrict the entries
- Edit Page 5
- Under Page Rendering section region, select REPORT and click
- Under Column Attributes
- Against REPORT_TYPE aliases
- click edit attributes left side of the column icon with
paper, pencil will be available.
- select LIST OF VALUES
- Under Named_LOV select LOV_LEDGER_TYPE
this has already been created by us
click APPLYCHANGES
- select Tabular Form Element
- In display as column select
Select List (Named LOV)
click APPLYCHANGES and RUN



Now you can see all the required changes on Page-5 Tabular Form (Ledger_Master)

Now, our next task creation of Master – Detail Form for Voucher Entries.

36-HOW TO CREATE MASTER-DETAIL FORM

→ Application Builder

→ Select Application

→ Edit Page 6 Ledger_Master

→ Page Rendering

→ Region section click create Button

→ select Form Click NEXT

→ select Master – Detail Form Click NEXT

DEFINE MASTER TABLE

→ Leave owner default

→ select Table Name as Raak_Trans_Master (all the columns will get displayed in available columns, using '>>' icon shift all the columns to displayed columns click NEXT

DEFINE DETAIL TABLE

→ Leave default Yes against Show only related tables

→ Leave the default owner against Table Owner

→ select the Raak_Trans_Detail against Table Name will display all the columns in available columns and push all the columns to displayed column box click NEXT

DEFINE PRIMARY KEY

→ leave the selected default values click NEXT

LINK MASTER and DETAIL

→ Based on foreign key establishment made already under table attributes,
the link will be displayed and leave the default and click NEXT

MASTER TABLE PRIMARY KEY DEFAULT

→ select existing trigger and click NEXT

DETAIL TABLE PRIMARY KEY DEFAULT

→ select existing trigger and click NEXT

DEFINE MASTER OPTIONS

→ leave the default values and click NEXT

CHOOSE LAYOUT

→ select Edit detail as tabular form on same page and click NEXT

PAGE ATTRIBUTES

→ Page Title for page 7 may be modified as Voucher_details

→ Under breadcrumb entry select parent entry as Transaction_menu

Click

NEXT

IDENTIFY TABS

select Use an existing tab set and reuse an existing tab within that tabset

select Transaction_Menu under tabset click NEXT

Identify the Vouchers as tab click NEXT

allow the default button labels and click NEXT

allow the default branching page click NEXT

click CREATE under Master Detail Confirmation

Linking this page No 6 to Page 3

Since Transaction_Menu exists in Page No.3 and clicking Vouchers tab control should be shifted to Page No.6 (Master Detail form for Vouchers)

Edit Page No:3

→ Shared Components → Tabs → select Vouchers tab

→ Under current for Pages select the Tab Page as 6 instead of existing page 3 (made at the time of creation temporarily) Click APPLY

CHANGES

Run the page 3 click the Voucher Tab will take you to Page 6. Play with data and click Return 2 Page will take you to Main Menu.

At this stage we have to attach some ornamental attributes, like beautification and controls to these master-details forms, (Page-6 & Page-7)

In PAGE – 7 Voucher_Master

(a) Voucher_type should have select list (static values generated through

LOV.

→ Edit Page 7

→ click p7_voucher_type (Column Page Rendering Section Item)

→ click LOV

→ click create or edit static values

→ Display Value	Return Value
Payment Voucher	P
Receipt Voucher	R

Journal J

→ Apply changes

goto Name Section

→ select selectList against Display as

→ ApplyChanges → RUN

(b) Voucher_Number should not be editable / enterable one display

→ click p7_voucher_number (Column Page Rendering Section Item)

→ select selectList against Display as

→ ApplyChanges → RUN

(c) Voucher_date must have date picker

→ click p7_voucher_date (Column Page Rendering Section Item)

→ select Date Picker (Use application date format mask)

→ ApplyChanges → RUN

(d) Voucher_Naration should have more space or width should be increased.

(e) Debit_Total, Credit_Total must be display field with formatted output.

→ click p7_debit_total (Column Page Rendering Section Item)

→ select selectList against Display as

→ ApplyChanges → RUN

(f) Debit_total, Credit_total must display the totals of voucher_detail table debit, credit.

→ click p7_debit_total (Column Page Rendering Section Item)

→ select selectList against Display as

→ ApplyChanges → RUN

In Page-7 (Voucher_Detail)

(a) Ledger_Code must be only for display

→ Edit Page 7

→ In the section Region Raak_Master_Detail Report will be available. Click report

→ Under column attributes section, click edit icon of ledger_code

→ Under Tabular Form Element section display as column will have TEXT as value. Change it as Display as Text.

→ Click ApplyCHANGES and RUN

(b) Ledger_name must have POPUP for selection and on selection of Ledger_Name through popup, Ledger_code should get populated. Ledger_name width should be increased.

→ Edit Page 7

→ In the section Region Raak_Master_Detail Report will be available. Click report

→ Under column attributes section, click edit icon of ledger_name

→ Under Tabular Form Element section display as column may be

filled with PopUP LOV (named LOV)

- Element Width may be fixed as 50
- Under List of Values section Named LOV column may be filled as
LOV_LEDGER_NAMES click applyCHANGES and RUN

(c) debit column decimal precision should be 2 with comma format.

- In the section Region Raak_Master_Detail Report will be available.

Click report

- Under column attributes section, click edit icon of debit
- Under column Definition
- say yes to compute sum
- Heading alignment as right
- column alignment as right
- Under column Formatting
- Number format as 99G99G990D00

(d) Credit column decimal precision should be 2 with comma format.

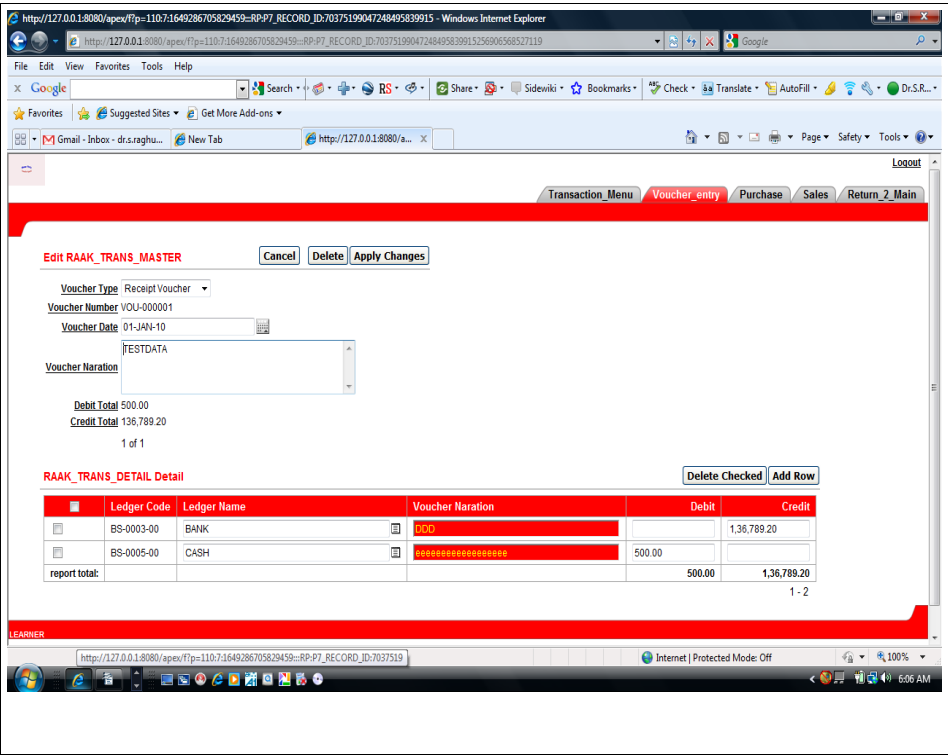
- In the section Region Raak_Master_Detail Report will be

available. Click report

- Under column attributes section, click edit icon of debit
- Under column Definition
- say yes to compute sum
- Heading alignment as right
- column alignment as right

- Under column Formatting
- Number format as 99G99G990D00

After affecting all the properties, the screen looks like as follows:



37 TEST DATA FOR MASTER-DETAIL FORM

The need of the test data is testing the master detail form and conceiving business logic for deriving control reports like trial balance, profit and loss account and balance sheet. Further while testing the form, we may require some minimal sophistications on data entry. Since we do not know what to debit or what to credit, we shall borrow the customer's data as such and use it.

Transaction No: 001

Introducing money as capital into the business			
Date	Account Head	Debit	Credit
'01-apr-2010	Capital		1000000
	Cash	100000	
	Bank	900000	

Transaction No: 002

Buying Machinery			
Date	Account Head	Debit	Credit
'02-apr-2010	Fixed Assets	500000	
	Bank		500000

Transaction No: 003

Purchase of Raw Material (partial credit and partial cheque payment)
--

Date	Account Head	Debit	Credit
'03-apr-2010	Purchase	500000	
	Bank		200000
	Xxx LTD		300000

Transaction No: 004

Salary / wages paid			
Date	Account Head	Debit	Credit
'04-apr-2010	Salary	10000	
	Cash		10000

Transaction No: 005

Manufacturing expenses paid partially			
Date	Account Head	Debit	Credit
'05-apr-2010	Expenses	35000	
	Cash		10000
	Bank		10000
	Payable-others		15000

Transaction No: 006

Sales			
Date	Account Head	Debit	Credit
'06-apr-2010	Sales		800000
	Cash	100000	
	Bank	200000	
	Receivable- others	500000	

On clicking Voucher in Transaction Menu, control has been shifted to page number 6 and clicked CREATE.

Meanwhile, the above mentioned ledger_names should have been created in ledger_master, if not enter the following in ledger_master

Ledger_names	Report_type
CASH	BALANCE SHEET ITEM
BANK	BALANCE SHEET ITEM
CAPITAL	BALANCE SHEET ITEM
FIXED ASSETS	BALANCE SHEET ITEM
PAYBLES-OTHERS	BALANCE SHEET ITEM
RECEIVABLE-OTHERS	BALANCE SHEET ITEM
PURCHASE	PROFIT LOSS ITEM
SALES	PROFIT LOSS ITEM
SALARY	PROFIT LOSS ITEM
EXPENSES	PROFIT AND LOSS ITEM

38-HOW TO CREATE REPORT PAGE

Application Builder

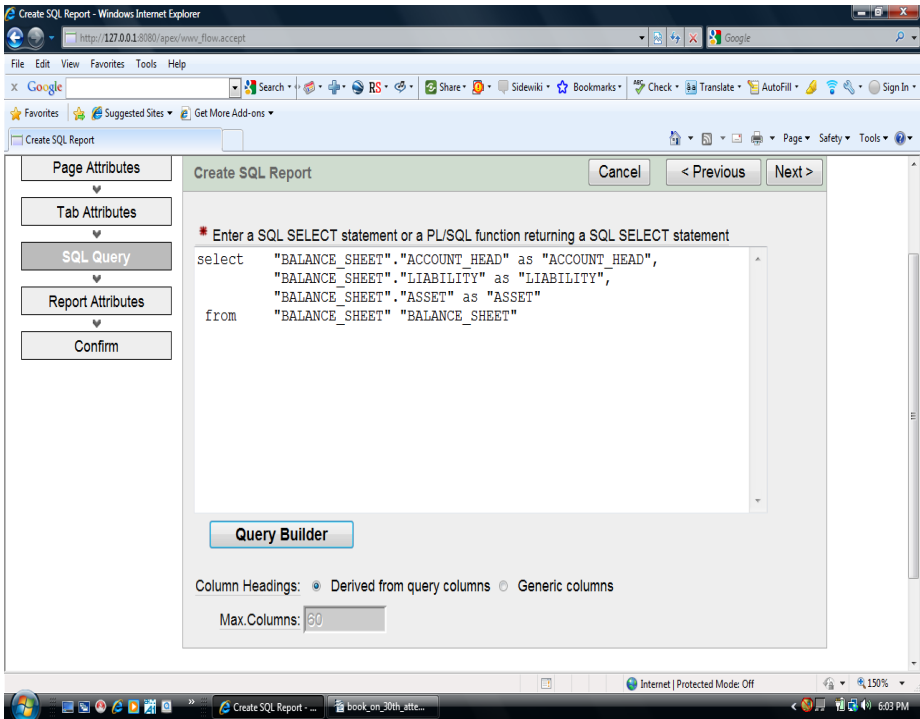
- select application
- create Page
- select report click NEXT
- select SQL Report click NEXT
- type report name as control reports
- select breadcrump
- select parent entry as report_menu click NEXT

Under Tab (Optional)

- select Use an existing tab set and reuse an existing tab within that tabset
- select Report_Menu under tabset click NEXT
- Identify the Control_Report as tab click NEXT

- allow the default for report attributes page click NEXT
- click FINISH under SQL Report Confirmaton

Linking this page No 8 to Page 4



Since Report Menu exists in Page No.4 and clicking Control Report tab control should be shifted to Page No.8 (SQL Report Page)

Edit page 8

→ Under region Report Title change it as Balance Sheet instead of Report 1.

control_reports - Windows Internet Explorer

http://127.0.0.1:8080/apex/f?p=112&601671294378856:::

File Edit View Favorites Tools Help

Google Search

control_reports

Logout

Reports_Menu Control_Reports Return 2 Main Menu

Reports_menu > control_reports

BALANCE SHEET

ACCOUNT HEAD	LIABILITY	ASSET
ACCUMULATED PROFIT	255000	
CAPITAL ACCOUNT	1000000	
CASH ACCOUNT		280000
FIXED ASSET		500000
PAYABLES - (SUPPLIER)	300000	
PAYABLES-OTHERS	15000	
RECEIVABLES - (CUSTOMERS)		500000
STATE BANK OF INDIA		290000
report total:	1570000	1570000

1 - 8

LEARNER

Done

Internet | Protected Mode: Off

150%

6:35 PM

→ Under Region select Report Under column attributes, select the column Liability and Asset and corresponding sum box may be ticked and click apply changes.

Now we require another report Profit and Loss Account in the same page but under different region. Shall we try?

Edit Page – 8

→ Under region click create button

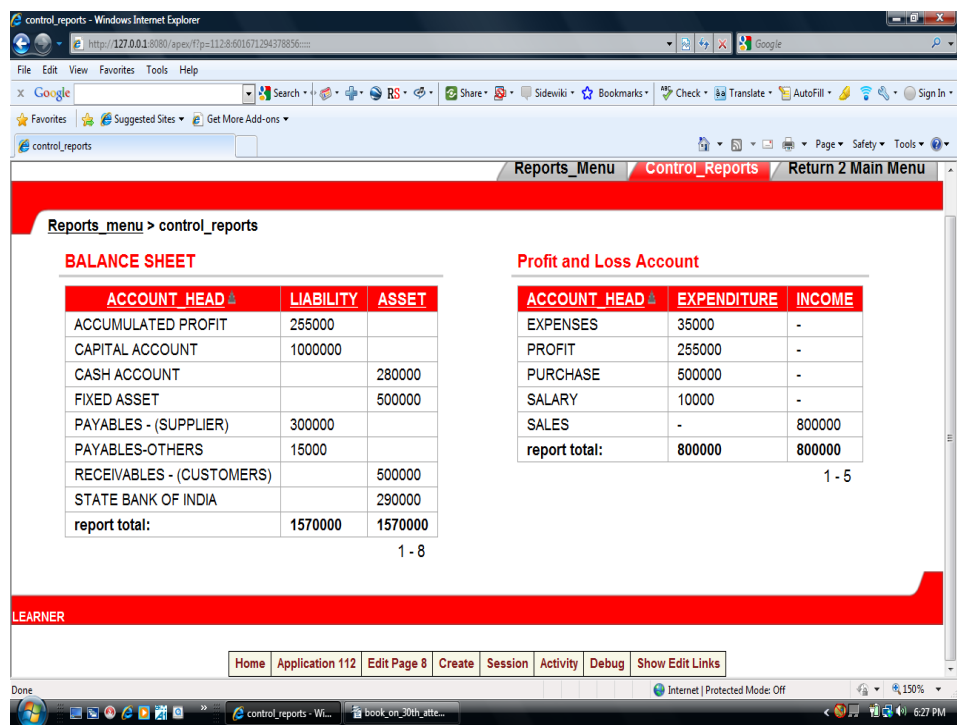
→ select report and click next

→ select SQL Report and click NEXT

→ Under Title Profit and Loss Account type and click NEXT

→ type

select "PROFIT_LOSS"."ACCOUNT_HEAD" as "ACCOUNT_HEAD",
"PROFIT_LOSS"."EXPENDITURE" as "EXPENDITURE",
"PROFIT_LOSS"."INCOME" as "INCOME"
from "PROFIT_LOSS" "PROFIT_LOSS"
→ click NEXT
→ Leave default in report attributes
→ allow default on conditional display click CREATE REGION



As such we have created 8 pages, almost miniature accounting application has been created. At this stage, you will gain enormous confidence that any layman too can create application software. This confidence can be cashed towards professionalization and upliftment on further sophistication on application software.

39 – HOW TO RUN APPLICATION

Login to apex using

workspace learner

schema learner

password learner

==> select the application builder

==> select the application and double click

==> give login credentials

your browser will display the main page.

Right click on the mouse

select the option for creating shortcut and placing the
shortcut on desktop

job ends.

Whenever, we double click the icon available in the desktop,
you shall land into your application.

Alternatively, you shall give the application address created in
your browser like

<http://127.0.0.1:8080/apex/f?p=110:1:3765189811749977:::>

40 – HOW TO THANK YOU

I DO NOT FIND RIGHT WORDS TO THANK READERS .