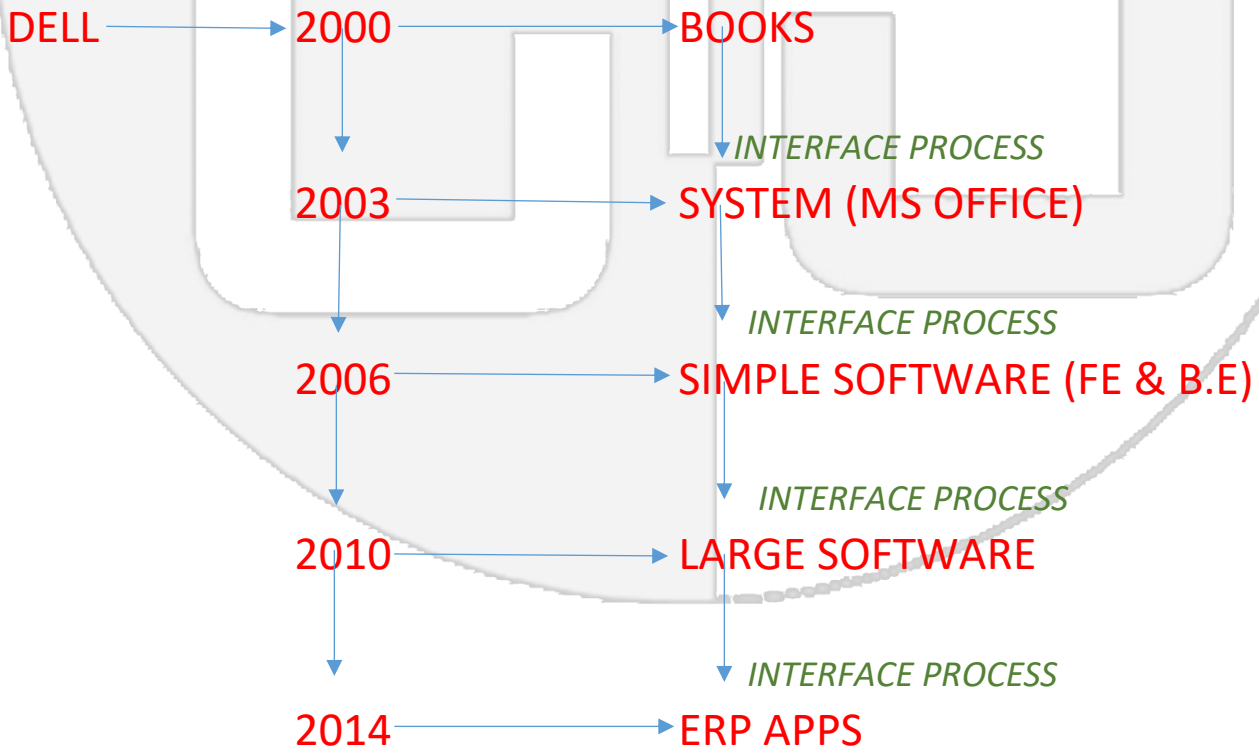
CONVERSIONS AND INTERFACES

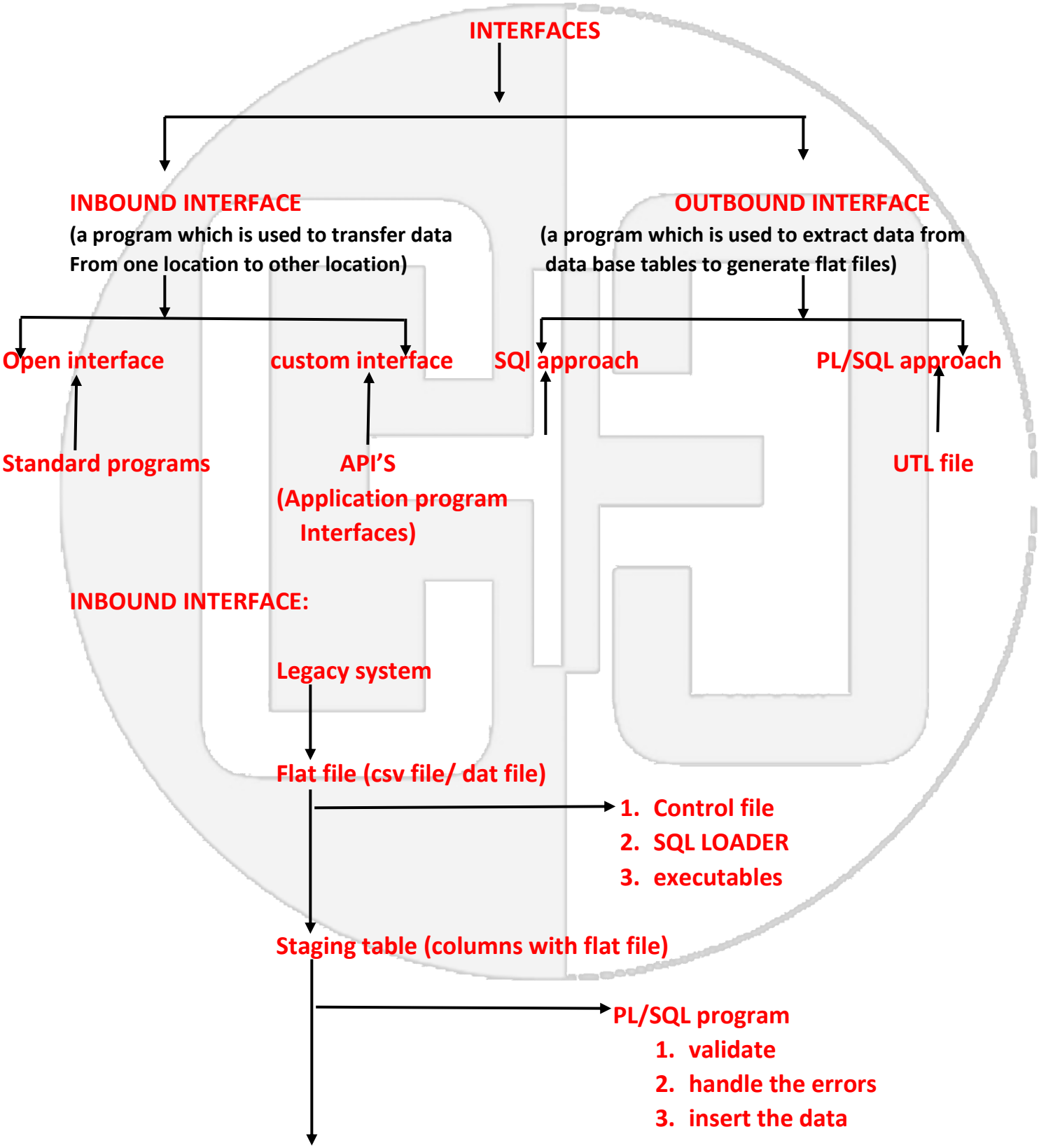
1. Introduction:

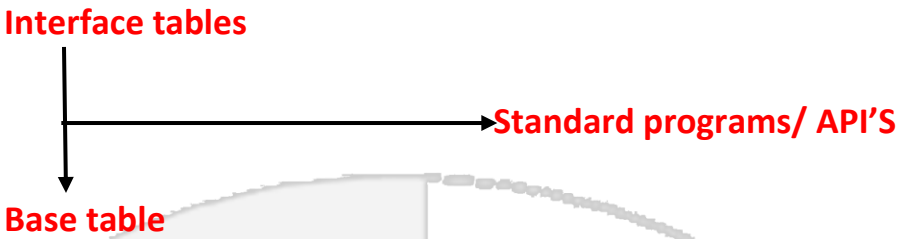
a. **Inbound interface is used to load the data into oracle apps from other party systems (external system/ legacy system). Here data should be formatted according to the target system and then move into application by using open interface or API**

Similar to inbound we can see the conversion but the only diff b/w two cases is only few

- **Conversion—it is one time process to load the entire data, here we can load batch wise also. But once we finish the data conversion we no need to process again and again.**
- **Inbound interface: is continuous data loading into oracle apps from other system, it is scheduled process in organization**
- **In case of conversion and interface we can use sql*loader, UTL_FILES. We can also use external table approach, sometimes db_links as well.**
- **Interface is different in way of getting data. After few days we have to delta delete to remove old data**







sql*loader:

it is the default tool in oracle apps to load the data into some tables.

The sql*loader control file:

The control file provides the fallowing information to the sql*loader

--the name and location of the input data file.

--the format of the record of the input of data file.

--the name of the tables or tables to be loaded

--selection criteria defining which records from the input file contains data to be inserted to the destination database tables.

--the name and location of the bad file and discard file, control file extension is (.ctl)

Control file syntax:

```
Options(skip=1)
Load data
Infile '&1'/data file path
Replace/ append/ insert/ truncate
Into table staging_table_name
Fields terminated by ','
Optionally enclosed by "
Trailing nullcals
Column1, column2
---------------------
Who colums
)
```

SKIP:

Skip is used to ignore the entire record while loading if we take skip=1, it would not allow the first record/header record (i.e., column heading).

This command will not be helpful to skip the bottom level records and we cannot skip specific record

DATAFILE

- **Inplace of data file path, we can give hard coded path like below.**
  C:/interface/inv_data.txt
  Or
  '&1'
- **Here '&' is used to pass the parameter as file path, we should use only numeric value after the "&"**

---taking parameter is the most preferable method.

INSERT:

- **To use this keyword, table should not have single record**

APPEND

- **To add new records to the table and will allow to insert records**

REPLACE

- **It will over write the records if already exists**

FIELS TERMINATE BY:

- **It is a column separator inside the data file**
- **Eg: incase of data file, if the fields are separated with ' , ' like below**

  Invoice_id, invoice_type, vendor_name, 6019, standard, 5033

  Then we should write the syntax like
  Fields terminated by ' , '

  Inplace of ',' , sometimes we can also use the pipe (||) symbol, negation (~)
  These operators depends upon the external system reporting method.

OPTIONALLY ENCLOSED BY:

This is used to accept the entire string with space

TRAILING NULL CALS:

It allows null values from data file.

FILLER

To ignore the entire column while loading

Note: while writing code for control file we can use some third party editors like notepad ++, text pad.

To use text pad for oracle we need to download oracle/sql/plsql syntax file and place it into c:/program file/textpad-6/system/sql.syn

Open the text pad, to high light the sql

Create executable by taking sql*loader

Create concurrent program

Attch to your responsibility

Submit it from the responsibility

It will generate 3 files bad file, log file, discard file

Bad file:

The records having datatype mismatch and column mismatch

Discard file:

When condition failed records

Log file:

Complete info like

how many records loaded, rejected,

what was cpu time, execution start time, end time.

Staging table name, column name

TOP 5 CONVERSIONS:

1. VENDOR INTERFACE:

This interface is used to import suppliers, supplier sites and site contacts into Oracle applications.

INTERFACE TABLES	BASE TABLES	INTERFACE PROGRAMS	VALIDATIONS
AP_SUPPLIERS_INT AP_SUPPLIER_SITES_INT AP_SUP_SITE_CONTACT_INT **AP_SUP_INT_REJECTIONS**	PO_VENDORS PO_VENDOR_SITES_ALL PO_VENDOR_CONTACTS	Supplier Open Interface Import Supplier Sites Open Interface Import Supplier Site Contacts Open Interface Import	• Check if vendor name, number, type already exists • Check if vendor site already exists(operating unit, country code, invoice currency code, payment currency) • Check if site contact already exists (first name, last name)

2. AP INVOICE:

This interface helps us to import vendor invoices into Oracle applications from external systems into Oracle Applications

Interface tables	Base tables	Concurrent program	validations
AP_INVOICES_INTERFACE AP_INVOICE_LINES_INTERFACE **AP_INTERFACE_REJECTIONS**	AP_INVOICES_ALL-header information AP_INVOICE_DISTRIBUTIONS_ALL – lines info	Payables Open Interface Import	• Check for valid vendor • Check for Source, Location, org_id, currency_code's validity • Check for valid vendor site code. • Check if record already exists in payables interface table.

3. RECEIVING:

The Receiving Open Interface is used for processing and validating receipt data that comes from sources other than the Receipts window in Purchasing.

Interface tables	Base tables	Concurrent program	validations
RCV_HEADERS_INTERFACE RCV_TRANSACTIONS_INTERFACE **PO_INTERFACE_ERRORS**	RCV_SHIPMENT_HEADERS RCV_SHIPMENT_LINES RCV_TRANSACTIONS	RECEIVING OPEN INTERFACE	• Check that SHIPPED_DATE should not be later than today. • Check if vendor is valid. • If Invoice number is passed, check for its validity • Check if Item is valid.

4. GL budget interface

Budget interface lets you load budget data from external sources into Oracle Applications.

Interface tables	Base tables	Concurrent program	validations
GL_BUDGET_INTERFACE	GL_BUDGETS GL_BUDGET_ASSIGNMENTS GL_BUDGET_TYPES	Budget Upload	• Check if CURRENCY_CODE is valid. • Check if SET_OF_BOOKS_ID is valid. • Check if BUDGET_ENTITY_NAME (budget organization) is valid.

5. GL INTERFACE:

This interface lets you import journals from other applications like Receivables, Payables etc to integrate the information with General Ledger.

INTERFACE TABLES	BASE TABLES	CONCURRENT PROGRAM	VALIDATIONS

GL_INTERFACE	GL_JE_HEADERS
	GL_JE_LINES
	GL_JE_BACTHES
		Journal Import
		Journal Posting — populates GL_BALANCES
			• Validate SOB, journal source name, journal category name, actual flag
			  ○ A – Actual amounts
			  ○ B – Budget amounts
			  ○ E – Encumbrance amount
			• If you enter E in the interface table, then enter appropriate encumbrance ID, if B enter budget id.
			• Check if accounting date or GL date based period name is valid (i.e., not closed).
			• Check if accounting date falls in open or future open period status.
			• Check chart of accounts id based on Sob id.
			• Check if code combination is valid and enabled.
			• Check if record already exists in GL interface table.
			• Check if already journal exists in GL application.