

# INF20010 / 60014 Assignment 2 (version 1.1)

Assignment Value: 15% of your final mark

The assignment is to be done groups of up to 4 people

Due Date/Time: 11:59pm Friday 30<sup>th</sup> May 2014

## SUBMISSION REQUIREMENTS

All submissions must be made by a team using ESP via: <https://esp.ict.swin.edu.au/>.

### Team Registration

- You must register your team in ESP BEFORE by 4pm Friday 16 May 2014.
- You cannot submit your assignment unless you join an ESP team.
- If you do not have a team by this date you will incur a penalty.

### Files required

- Your submission must be in a single ZIP file.
- The .ZIP file must contain: Ass2\_Script.TXT and Ass2\_Output.TXT

## GENERAL DESCRIPTION:

The aim of this assignment is to use ETL to populate a Data Warehouse by:

- Extracting data from source tables.
- Transforming and cleaning the data, logging all changes.
- Loading the data into a data warehouse.

Finally you will be required to write queries based on data stored in the data warehouse.

Note: All operations of ETL must be achievable by running SQL scripts. This scripts must work on any source data values. (If the convener decides to change the source data values, then your code must still work correctly. You cannot assume that a particular data value such as person Fred Smith or that Software Product 15 will always exist in the source data tables.)

## SOURCE DATA BACKGROUND:

HARRIS and RIGG were two medium sized organizations located in Melbourne and Brisbane. Both companies imported and sold similar products. Both companies sold their products via their on-line stores.

Customers for both companies came from all around Australia. Most customers remained loyal to the company that they first purchased items from. So generally HARRIS customers didn't buy from RIGG and RIGG customers didn't buy from HARRIS. However there are some exceptions.

HARRIS and RIGG have now merged and become the ACME organization. The product data have been merged and are now common to both branches of ACME. The customer and sales data have **not** been merged and are still stored separately within each branch. This means that there are two different customer tables and two different sales tables.

The convenor of INF20010/60014 has a copy of all source tables in his database account. Read-Only public access has been defined so you can refer to these tables. You cannot modify data in these tables. There is no need to copy the data in these tables to your account. Note: The data in the source tables may change!

**SOURCE DATA TABLES:**

These are the names of the existing source tables:

A2PRODUCT	A2_Tnn_SALEMELB	A2CUSTCATEGORY
A2_Tnn_CUSTBRIS	A2MANUFACTURER	A2DATEDATA
A2_Tnn_CUSTMELB	A2SHIPPING	
A2_Tnn_SALEBRIS	A2PRODCATEGORY	

Where tables contain Tnn, use your team number to complete the filename.

E.g. If you are in ESP Team 06, then you must use the table A2\_T06\_CUSTBRIS.

Source data from the source tables cannot be guaranteed to be clean.

Your task is to check data in the source tables.

- Some rows will be no errors
- Some rows may have errors that can be fixed
- Some rows may have errors that cannot be fixed and will be rejected

Rows that have no errors or that can be fixed will be uploaded into Data Warehouse etables

**ERROR EVENT TABLE**

During the ETL process, you must keep track of which source data rows need to be fixed or rejected.

The schema for this table is:

```

ERROREVENT (
ERRORID          INTEGER,
SOURCE_ROWID    ROWID,
SOURCE_TABLE    VARCHAR2(30),
FILTERID         INTEGER,
DATETIME         DATE,
ACTION           VARCHAR2(6),
CONSTRAINT ERROREVENTACTION
CHECK (ACTION IN ('SKIP','MODIFY'))
);

```

**DATA WAREHOUSE TABLES:**

You will create some tables in your account that simulate a Data Warehouse

These are the names of the data warehouse tables that you must create:

DWPROD	DWCUST
DWDATEDATA	DWSALE

**PART 1 PRODUCT DATA (20%)**

- a) Write code to create the ErrorEvent table DDL and place the code into the Ass2\_Script file.

Add appropriate Drop Table and Create Sequence statements for this table in the appropriate section of the script file so that each time you run the script the code can be tested without any effects from previous attempts

- b) Write the stored procedure named SP\_CLEAN\_PRODUCT. This code must apply the filters listed below to the source product table and update the ERROREVENT table where appropriate

Filter No.	Quality Check Issue	Transformation Action Required
1.	Product Category is does not match a PK value in the Product Category table. <ul style="list-style-type: none"> <li>Set action to Modify</li> </ul> Note: The source tables do <b>not</b> contain any Foreign Key constraints	Set the PRODCATNAME to 'UNKOWN' during the upload process.

- Each new row added to the ERROREVENT table must be given unique ERRORID value. The ERRORID value must be obtained from a sequence named ERROREVENTSEQ.
- The SOURCE\_ROWID value must be set to the ROWID of the offending source data row
- The SOURCE\_TABLE value must be set to the table name 'A2PRODUCT'
- The FILTERID value must be set to the appropriate filter number
- The DATETIME value must be set to current value of SYSDATE
- The ACTION value must be set to either 'MODIFY' or 'SKIP'

- c) Write code to create the DWPROD table DDL and place the code into the Ass2\_Script file.

The schema is DWPROD(DWPRODID, DWSOURCETABLE, DWSOURCEID, PRODNAME, PRODCATNAME, PRODSHIPNAME, PRODMANUNAME, PRODSHIPNAME)

Add appropriate Drop Table and Create Sequence statements for this table in the appropriate section of the script file so that each time you run the script the code can be tested without any effects from previous attempts

- d) Write the stored procedure named SP\_UPLOAD\_PRODUCT. Place the SP code into the script file. This code uploads data from the source product table into the DWPROD table.
- All source rows not referenced in the ErrorEvent table are to be uploaded to DWPROD
  - Each new product added to the DWPROD table must be given unique DWPRODID value. The DWPRODID value must be obtained from a sequence named DWPRODSEQ.
  - The DWSOURCETABLE value must be set to 'A2PRODUCT'.
  - The DWSOURCEID value must be set to the customer id of the source table
  - All source rows referenced as MODIFY in the ErrorEvent table must adjust the data values as they are transferred to the DWPROD table. (You must not attempt to update the source table values).
  - All source rows referenced as SKIP in the ErrorEvent table must not be uploaded to the DWPROD table
  - Data associated with each product is located in the tables named A2PRODCATEGORY, A2MANUFACTURER, A2SHIPPING. The data from these tables must be denormalised so that it is all uploaded data is contained in the single table named DW\_PROD.
- e) Write an anonymous block that executes SP\_CLEAN\_PRODUCT and SP\_UPLOAD\_PRODUCT. Place the code into the script file
- f) Modify the SQL statement for QUERY1 (in the script file) so that it matches the table names and column names used in your database. This query generates a summary of data stored in the ERROREVENT table.

Note: There is no need to modify query 2. Query 2 will count the number of rows in each of the DW tables

**PART 2 BRISBANE CUSTOMER DATA (20%)**

- a) Write the stored procedure named SP\_CLEAN\_CUST\_BRIS. This code must apply the filters listed below to the source customer table and update the ERROREVENT table where appropriate

Filter No.	Quality Check Issue	Transformation Action Required																				
2.	<p>CustCategory does not match a PK value in the CUSTCATEGORY source table.</p> <ul style="list-style-type: none"><li>Set action to Modify</li></ul>	Set the CUSTCATNAME to 'UNKOWN' during the upload process.																				
3.	<p>Customer Phone Number has invalid characters.</p> <ul style="list-style-type: none"><li>Set action to Modify</li></ul> <p>An invalid phone number is one that contains a space or a hyphen</p> <p>Customer Phone Number with valid characters does not have length equal to 10.</p> <ul style="list-style-type: none"><li>Set action to Skip</li></ul>	Remove all invalid characters for rows that require modification during the upload process																				
4.	<p>Gender value is not M or F.</p> <ul style="list-style-type: none"><li>Set action to Modify</li></ul>	<p>For those rows that require modification do the following:</p> <p>If the gender value is one of the values in the GenderSpelling table then apply the correct spelling and upload (You are to create the GenderSpelling table)</p> <p>If the gender value is not one of the values in the GenderSpelling table then set the gender value to 'UNKOWN' and upload</p> <p>This is a list for the GenderSpelling Table</p> <table><tr><th>Invalid Value</th><th>New Value</th></tr><tr><td>MAIL</td><td>M</td></tr><tr><td>WOMAN</td><td>F</td></tr><tr><td>FEM</td><td>F</td></tr><tr><td>FEMALE</td><td>F</td></tr><tr><td>MALE</td><td>M</td></tr><tr><td>GENTLEMAN</td><td>M</td></tr><tr><td>MM</td><td>M</td></tr><tr><td>FF</td><td>F</td></tr><tr><td>FEMAIL</td><td>F</td></tr></table> <p>Treat lower case values as upper case.</p>	Invalid Value	New Value	MAIL	M	WOMAN	F	FEM	F	FEMALE	F	MALE	M	GENTLEMAN	M	MM	M	FF	F	FEMAIL	F
Invalid Value	New Value																					
MAIL	M																					
WOMAN	F																					
FEM	F																					
FEMALE	F																					
MALE	M																					
GENTLEMAN	M																					
MM	M																					
FF	F																					
FEMAIL	F																					

If you attempt Filter 4, add the Drop & Create GenderSpelling table code to GENDERSPELLING section of the script file. Also add all required INSERT statements to populate the table.

- b) Write code to create the DWCUST table DDL and place the code into the Ass2\_Script file.

The schema is DWCUST(DWCUSTID, DWSOURCEIDBRIS, DWSOURCEIDMELB, FIRSTNAME, SURNAME, GENDER, PHONE, POSTCODE, CITY, STATE, CUSTCATNAME)

Add appropriate Drop Table and Create Sequence statements for this table in the appropriate section of the script file so that each time you run the script the code can be tested without any effects from previous attempts

- c) Write the stored procedure named SP\_UPLOAD\_CUSTOMER\_BRIS. Place the SP code into the script file. This code uploads data from the source customer table into the DWCUST table.
- All rows not listed in the ErrorEvent table are to be uploaded to DWCUST
  - Each new customers added to the DWCUST table must be given unique DWCUSTID value. The DWCUSTID value must be obtained from a sequence named DWCUSTSEQ.
  - The DWSOURCEIDBRIS value must be set to the customer id of the source table
  - All source rows referenced as MODIFY in the ErrorEvent table must adjust the data values as they are transferred to the DWCUST table. (You must **not** attempt to update the source table values).
  - All source rows referenced as SKIP in the ErrorEvent table must not be uploaded to the DWCUST table
  - Data associated with each Brisbane customer is located in the tables named A2CUSTBRIS, A2CUSTCATEGORY. The data from these tables must be **denormalised** so that it is all uploaded data is contained in the single table named DW\_CUST.

### **PART 3 BRISBANE SALES DATA (20%)**

- a) Write the stored procedure named SP\_CLEAN\_SALES\_BRIS. This code must apply the filters listed below to the source sales table and update the ERROREVENT table where appropriate

Filter No.	Quality Check Issue	Transformation Action Required
5.	Product ID does not match a SourceProdID value in the DWPROD table where the SourceTableName is 'A2PRODUCT' <ul style="list-style-type: none"> <li>Set action to SKIP</li> </ul>	Ignore all SKIP rows during the upload process.
6.	Customer ID does not match a Source CustID value in the DWCUST table <ul style="list-style-type: none"> <li>Set action to SKIP</li> </ul>	Ignore all SKIP rows during the upload process.
7.	Shipdate is earlier than SaleDate <ul style="list-style-type: none"> <li>Set action to MODIFY</li> </ul>	Modify the Ship Date so that it is equal to the Sale Date + 2 days during the upload process.
8.	Sale Price is Null <ul style="list-style-type: none"> <li>Set action to MODIFY</li> </ul>	Modify the Sale Price so that it matches the most recent sale price for that product in the DWSALE table during the upload process.

- b) Write code to create the DWSALE table DDL and place the code into the Ass2\_Script file.

The schema is DWSALE(DWSALEID, DWCUSTID, DWPRODID, DWSOURCEIDBRIS, DWSOURCEIDMELB, QTY, SALEDATE, SHIPDATE, SALEPRICE)

Add appropriate Drop Table and Create Sequence statements for this table in the appropriate section of the script file so that each time you run the script the code can be tested without any effects from previous attempts

- c) Write the stored procedure named SP\_UPLOAD\_SALE\_BRIS. Place the SP code into the script file. This code uploads data from the source sale table into the DWSALE table.
- All rows not listed in the ERROREVENT table are to be uploaded to DWSALE
  - Each new sale added to the DWSALE table must be given unique DWSALEID value. The DWSALEID value must be obtained from a sequence named DWSALESEQ.
  - The DWSOURCEIDBRIS value must be set to the sale id of the source table
  - The DWCUSTID value must be set to the appropriate DWCUSTID of the DWCUST table
  - The DWPRODID value must be set to the appropriate DWPRODID of the DWPROD table
  - All source rows referenced as MODIFY in the ErrorEvent table must adjust the data values as they are transferred to the DWSALE table. (You must **not** attempt to update the source table values).
  - All source rows referenced as SKIP in the ErrorEvent table must not be uploaded to the DWSALE table

#### **PART 4 MELBOURNE CUSTOMER DATA (10%)**

- a) Normally, the Melbourne Customers would require the same checking as the Brisbane customers. However in this assignment you may assume that all of the data values in this table are clean. So do not test for filters 2,3 & 4.
- b) Write the stored procedure named SP\_CLEAN\_CUST\_MELB. This code must apply the filters listed below to the source customer table and update the ERROREVENT table where appropriate

Filter No.	Quality Check Issue	Transformation Action Required
9.	The customer name and phone number matches details in the DWCUST table. <ul style="list-style-type: none"> <li>Set action to Modify</li> </ul>	Do not add a new row to the DWCUST table. Instead update the DWSOURCEIDMELB value to be the source customer id

- c) Write the stored procedure named SP\_UPLOAD\_CUSTOMER\_MELB. Place the SP code into the script file. This code uploads data from the source customer table into the DWCUST table.
  - All rows not listed in the ERROREVENT table are to be uploaded to DWCUST
  - Data associated with each Melbourne customer is located in the tables named A2CUSTMELB, A2CUSTCATEGORY. The data from these tables must be **denormalised** so that it is all uploaded data is contained in the single table named DW\_CUST.



### **PART 5 MELBOURNE SALES DATA (10%)**

- a) Write the stored procedure named SP\_CLEAN\_SALES\_MELB. This code must apply the filters listed below to the source product table and update the ErrorEvent table where appropriate

Code in this SP may use code that is almost identical to filters above. While this is not an ideal way to write code, it is OK for this assignment. It also makes marking of the assignment more simple. ☺

Filter No.	Quality Check Issue	Transformation Action Required
10.	Product ID does not match a SourceProdID value in the DWPROD table where the SourceTableName is 'A2PRODUCT'  • Set action to SKIP	Ignore all SKIP rows during the upload process.
11.	Customer ID does not match a Source CustID value in the DWCUST table  • Set action to SKIP	Ignore all SKIP rows during the upload process.
12.	Shipdate is earlier than SaleDate  • Set action to MODIFY	Modify the Ship Date so that it is equal to the Sale Date + 2 days during the upload process.
13.	Sale Price is Null  • Set action to MODIFY	Modify the Sale Price so that it matches the most recent sale price for that product in the DWSALE table during the upload process.

- b) Write the stored procedure named SP\_UPLOAD\_SALE\_MELB. Place the SP code into the script file. This code uploads data from the source sale table into the DWSALE table.
- All rows not listed in the ErrorEvent table are to be uploaded to DWSALE
  - Each new sale added to the DWSALE table must be given unique DWSALEID value. The DWSALEID value must be obtained from a sequence named DWSALESEQ.
  - The DWSOURCEIDMELB value must be set to the sale id of the source table
  - The DWCUSTID value must be set to the appropriate DWCUSTID of the DWCUST table
  - The DWPRODID value must be set to the appropriate DWPRODID of the DWPROD table
  - All source rows referenced as MODIFY in the ErrorEvent table must adjust the data values as they are transferred to the DWSALE table. (You must **not** attempt to update the source table values).
  - All source rows referenced as SKIP in the ErrorEvent table must not be uploaded to the DWSALE table

**PART 6 QUERIES (10%):**

Write SQL queries based on your data warehouse to display the following information.

Note: The values used to create these examples will not match current values in the database tables.

1. Adjust existing Query in Script file to match table and column names used
2. No changes required to existing Query in Script file
3. List each weekday (e.g. Monday, Tuesday...) and the total sales (qty \* saleprice) for that day.  
The list must be in descending total sequence. Note: 7 rows should be listed by this query

E.g.

WEEKDAY	TOTAL SALES
FRIDAY	52400
MONDAY	47340
SUNDAY	46235 ...

4. List each customer category and the total sales (qty \* saleprice).  
The list must be in ascending total sequence.

E.g.

CUSTCATNAME	TOTAL SALES
Exclusive	58200
Gold	61430
Silver	73439 ...

5. List each product manufacturer and the total qty sold  
The list must be in descending total sequence.

E.g.

PRODMANUNAME	TOTAL QTY SOLD
Lum-Tech	3210
Island Group	2956...

6. List the **top 10** customers based on total sales (qty \* saleprice).  
The list must be in descending total sequence.

E.g.

DWCUSTID	FIRSTNAME	SURNAME	TOTAL SALES
1045	Fred	Smith	2400
7776	Sue	Davies	2260
56	Emma	Jones	1998 ...

7. List the **bottom 10** products based on total qty sales  
The list must be in ascending total sequence.

E.g.

DWPRODID	PRODNAME	TOTAL SALES
321	Product BD	65
95	Product DY	89
206	Product JJ	132 ...

8. List the top city name based on total sales (qty \* price) for each state  
The list must be in ascending state sequence.

E.g.

STATE	CITY	TOTAL SALES
ACT	DOWNER	32010
NSW	TURRAMURRA	23400
QLD	HEATHWOOD	14200 ...

## **REQUIREMENTS**


### **The ASS2\_SCRIPT.TXT File**

Download the file ASS2\_SCRIPT.TXT from blackboard.

- Add your team's details and members to the top of this file.
- Add the code for each Part of this assignment to the script file.

#### **Testing the script.**

Before submitting the script file you must

- Copy the entire contents of the file and paste it into an SQL Developer worksheet.
- Press the Run script button 

The entire script should execute without error.

The script will drop tables and sequences, create tables and sequences, test ETL filters against source data, update the ErrorEvent table, upload data into the data warehouse tables and finally execute some SQL queries.

#### **Additional SPs & SFs**

If you write any additional SPs or SFs that are used in the ETL process, then you must include them in the script file. You will need to copy and format into the script file so that it obvious to your tutor.

It is important that you place the code for these SPs and SFs in the correct position in the script file. For instance if you call a SF before it has been created, then error messages will be generated and your script will not work.

Please test your script by doing the following:

- Make a backup of all your SPs and SFs
- Log into another team members account
- Drop all the SPs and SFs in that account
- Run the contents of the script file

### **THE ASS2\_OUTPUT.TXT file**

This file must contain the Script Output generated by the above script when executed in SQL Developer.

This should simply be a copy and paste action from the Script Output window in SQL Developer.