Data Warehouse Application (DWA)

The DWA application has been created to demonstrate an example front end for a Data warehouse. As a tutorial it assumes an understanding of the role of ETL (Extract Transform and Load) and reporting within a data warehouse.

The objective of this tutorial and application is:

1. To demonstrate a potential interface for the DW – ETL and reporting.
2. To demonstrate the use of packages, procedure and functions within an Apex Application.

The functionality of the DWA is:

1. To provide the ability to extract data from a new dataset and load it into the existing data warehouse. This involves:
   1. Performing a data quality check on the new data and logging any data errors
   2. Performing and transformations as necessary and logging any transformations
   3. Loading the ‘clean’ data into the DW (star schema model) tables

2. To provide the ability to ‘reset’ all the data in the system to as it was at the start of the tutorial. This is to help students using the application.

3. To provide a number of example data warehouse-type reports.

You should be able to run the setup script, create the DW\_ETL\_PKG, then import the apex application DW\_2016\_app. (this was done in Apex 5 – so there may be issues!)

Application Design

The data manipulation code will be contained within two packages:

1. The Data Ware house ETL package (DW\_ETL\_ pkg) – see below

This package consists of 3 procedures which relate the functionality in point 1 above:

Data\_quality\_check(), Data\_transform(), Data\_load

And

1. This is not done, you could do!

The Data Warehouse reports package (DW\_reports\_pkg)

This package contains 2 procedures for two different reports, they are:

New\_emps\_by\_year() and New\_emps\_in\_a\_year()

1. The ‘reset’ functionality will be included as code behind a button on the application.

-- DWA set up script

-- emp1 and emp2 represent new data for the DW

-- the DW tables are dept\_dim, time\_dim and fact\_emp

-- set up all the tables for the ETL

Drop table emp1;

Drop table emp2;

Drop table clean\_emp1;

Drop table clean\_emp2;

Drop table transform\_emp1;

Drop table transform\_emp2;

Drop table tmp\_load1;

Drop table tmp\_load2;

Create table emp1 as select \* from scott.emp;

Create table emp2 as select \* from emp1;

Create table clean\_emp1 as select \* from emp1;

Create table clean\_emp2 as select \* from emp1;

Create table transform\_emp1 as select \* from emp1;

Create table transform\_emp2 as select \* from emp1;

DELETE FROM emp2;

-- all the tables below are temporary tables to hold data through the transformations

DELETE FROM clean\_emp1;

DELETE FROM clean\_emp2;

DELETE FROM transform\_emp1;

DELETE FROM transform\_emp2;

CREATE TABLE tmp\_load1 (empno NUMBER(4), deptno NUMBER(4), which\_year NUMBER(4));

CREATE TABLE tmp\_load2 (year NUMBER(4), deptno NUMBER(4), no\_of\_emps NUMBER(4));

-- heres the different data for emp2

INSERT INTO emp2(

SELECT DI\_SEQ.nextval, ename, job, mgr, hiredate, '20000', '100', 2 FROM emp1 WHERE DEPTNO = '20');

INSERT INTO emp2(

SELECT DI\_SEQ.nextval, ename, job, mgr, hiredate,'20000', '100', 3 FROM emp1 WHERE DEPTNO = '30');

INSERT INTO emp2(

SELECT DI\_SEQ.nextval, ename, job, mgr, hiredate,'20000', '100', 1 FROM emp1 WHERE DEPTNO = '10');

UPDATE emp2 SET ename = NULL WHERE deptno = 2;

-- these are the DW - star schema tables

DROP table FACT\_emp CASCADE CONSTRAINTs;

DROP table dept\_dim CASCADE CONSTRAINTs;

DROP table time\_dim CASCADE CONSTRAINTs;

CREATE TABLE FACT\_emp(

fact\_id INTEGER NOT NULL,

deptno INTEGER,

time\_id INTEGER,

no\_employees INTEGER,

-- Specify the PRIMARY KEY constraint for table "FACT\_emp".

-- This indicates which attribute(s) uniquely identify each row of data.

PRIMARY KEY (fact\_id)

);

CREATE TABLE dept\_dim(

deptno INTEGER NOT NULL,

dept\_desc VARCHAR(10),

-- Specify the PRIMARY KEY constraint for table "dept\_dim".

-- This indicates which attribute(s) uniquely identify each row of data.

PRIMARY KEY (deptno)

);

CREATE TABLE time\_dim(

time\_id INTEGER NOT NULL,

time\_period INTEGER,

-- Specify the PRIMARY KEY constraint for table "time\_dim".

-- This indicates which attribute(s) uniquely identify each row of data.

PRIMARY KEY (time\_id)

);

--put some existing data into the dim tables

INSERT INTO dept\_dim (deptno, dept\_desc) SELECT deptno, dname FROM dept;

INSERT INTO time\_dim VALUES (1, 1978);

INSERT INTO time\_dim VALUES (2, 1979);

INSERT INTO time\_dim VALUES (3, 1980);

INSERT INTO time\_dim VALUES (4, 1981);

-- and fact

DROP sequence FACT\_SEQ;

create sequence FACT\_SEQ

start with 1

increment by 1

maxvalue 10000

minvalue 1;

INSERT INTO fact\_emp VALUES (fact\_seq.nextval, 10, 1, 3);

INSERT INTO fact\_emp VALUES (fact\_seq.nextval, 20, 1, 1);

INSERT INTO fact\_emp VALUES (fact\_seq.nextval, 30, 1, 4);

INSERT INTO fact\_emp VALUES (fact\_seq.nextval, 10, 2, 6);

INSERT INTO fact\_emp VALUES (fact\_seq.nextval, 20, 2, 4);

INSERT INTO fact\_emp VALUES (fact\_seq.nextval, 30, 2, 2);

INSERT INTO fact\_emp VALUES (fact\_seq.nextval, 40, 2, 6);

-- set up tables log the data errors

DROP TABLE Data\_Quality\_Errors;

CREATE TABLE Data\_Quality\_Errors

(error\_id integer NOT NULL,

empno NUMBER(4),

error\_desc VARCHAR(50),

status VARCHAR(10));

DROP sequence DQ\_SEQ;

create sequence DQ\_SEQ

start with 1

increment by 1

maxvalue 10000

minvalue 1;

-- and a table to record the transformations

DROP table Data\_transformations;

CREATE TABLE Data\_Transformations

(transform\_id NUMBER(4) NOT NULL,

table\_name VARCHAR2(30),

row\_id NUMBER(4),

data\_error\_code NUMBER(4),

transform\_desc VARCHAR(50),

transform\_date DATE,

transform\_status VARCHAR(10),

status\_update\_date DATE);

DROP sequence DI\_SEQ;

Create sequence DI\_SEQ MINVALUE 50 MAXVALUE 9999999 INCREMENT BY 1 START WITH 50;

--Set up trigger to record transformations

create or replace trigger trg\_transform\_emp

after update on transform\_emp2

for each row

begin

INSERT INTO data\_transformations

(transform\_id, table\_name, row\_id, data\_error\_code, transform\_desc, transform\_date, transform\_status, status\_update\_date)

VALUES

(DI\_SEQ.nextval, 'clean\_emp2', :OLD.empno, '0', 'transform deptno 1-2-3 to 10-20-30', SYSDATE, 'completed', SYSDATE);

end;

--Package code

create or replace package DW\_ETL\_PKG is

procedure Data\_quality\_check(pv\_rows OUT NUMBER);

procedure data\_transform(pv\_rows OUT NUMBER);

procedure data\_load(pv\_rows OUT NUMBER);

END DW\_ETL\_PKG;

create or replace package body DW\_ETL\_PKG AS

procedure data\_quality\_check (pv\_rows OUT NUMBER)

IS

BEGIN

pv\_rows := 0;

INSERT INTO data\_quality\_errors

SELECT DQ\_SEQ.nextval, emp1.empno, 'null values', 'not fixed'

FROM emp1 WHERE ename IS NULL

OR job IS NULL

OR hiredate IS NULL

OR SAL IS NULL

OR deptno IS NULL;

pv\_rows := TO\_CHAR(SQL%RowCount);

DELETE FROM clean\_emp1;

INSERT INTO clean\_emp1 (SELECT \* FROM emp1 WHERE

ename IS NOT NULL

AND job IS NOT NULL

AND hiredate IS NOT NULL

AND SAL IS NOT NULL

AND deptno IS NOT NULL);

INSERT INTO data\_quality\_errors

(SELECT DQ\_SEQ.nextval, emp2.empno, 'null values', 'not fixed'

FROM emp2 WHERE ename IS NULL

OR job IS NULL

OR hiredate IS NULL

OR SAL IS NULL

OR deptno IS NULL);

pv\_rows := pv\_rows + TO\_CHAR(SQL%RowCount);

DELETE FROM clean\_emp2;

INSERT INTO clean\_emp2 ( SELECT \* FROM emp2 WHERE

ename IS NOT NULL

AND job IS NOT NULL

AND hiredate IS NOT NULL

AND SAL IS NOT NULL

AND deptno IS NOT NULL);

END data\_quality\_check;

PROCEDURE data\_transform (pv\_rows OUT NUMBER) IS

BEGIN

pv\_rows :=0;

DELETE FROM transform\_emp1;

DELETE FROM transform\_emp2;

INSERT INTO transform\_emp1 SELECT \* FROM clean\_emp1;

INSERT INTO transform\_emp2 SELECT \* FROM clean\_emp2;

UPDATE transform\_emp2 SET deptno = 10 WHERE deptno = 1;

pv\_rows := TO\_CHAR(SQL%RowCount);

UPDATE transform\_emp2 SET deptno = 20 WHERE deptno = 2;

pv\_rows := pv\_rows + TO\_CHAR(SQL%RowCount);

UPDATE transform\_emp2 SET deptno = 30 WHERE deptno = 3;

pv\_rows := pv\_rows + TO\_CHAR(SQL%RowCount);

END data\_transform;

PROCEDURE data\_load (pv\_rows OUT NUMBER) IS

BEGIN

pv\_rows :=0;

INSERT INTO tmp\_load1 (SELECT

empno, deptno, to\_number(to\_char(transform\_emp1.hiredate,'YYYY')) FROM transform\_emp1);

INSERT INTO tmp\_load2 (SELECT

which\_year, deptno, COUNT(\*) no\_of\_emps FROM tmp\_load1 GROUP by which\_year, deptno);

INSERT INTO FACT\_emp (fact\_id, deptno, time\_id, no\_employees)

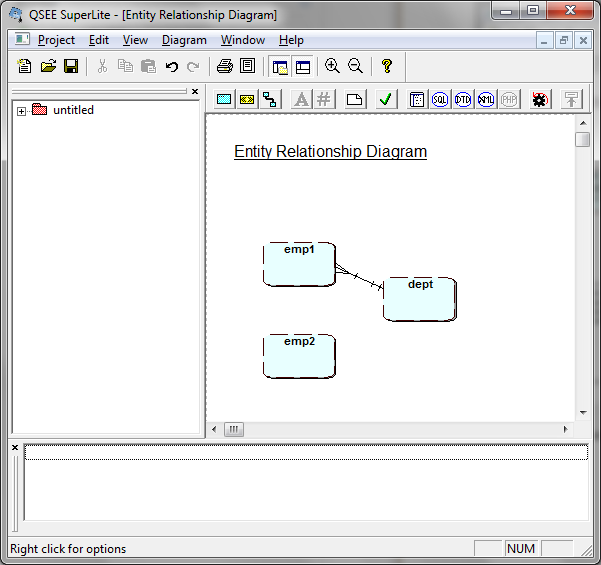
SELECT fact\_seq.nextval, deptno, time\_dim.time\_id, no\_of\_emps FROM tmp\_load2,time\_dim WHERE tmp\_load2.year = time\_dim.time\_period;

pv\_rows := TO\_CHAR(SQL%RowCount);

END data\_load;

END;

OLTP system



OLAP system

