|  |
| --- |
| CIS 229  Professor: Bob Desilets  Date 11/21/2018  Cedric L Mulumba  **Hands-On Assignments Part I** |
| **Assignment 8-1: Reviewing Dependency Information in the Data Dictionary** |
| desc user\_objects; |
|  |
| select rpad(object\_name, 23) "OBJECT\_NAME", status, timestamp  from user\_objects  where object\_type = 'PROCEDURE'; |
|  |
| desc user\_dependencies; |
|  |
| select name, type  from user\_dependencies  where referenced\_name = 'BB\_BASKET'; |
|  |
| **Assignment 8-2: Testing Dependencies on Stand-Alone Program Units** |
| CREATE OR REPLACE FUNCTION status\_desc\_sf  (p\_stage IN NUMBER)  RETURN VARCHAR2  IS  lv\_stage\_txt VARCHAR2(30);  BEGIN  IF p\_stage = 1 THEN  lv\_stage\_txt := 'Order Submitted';  ELSIF p\_stage = 2 THEN  lv\_stage\_txt := 'Accepted, sent to shipping';  ELSIF p\_stage = 3 THEN  lv\_stage\_txt := 'Backordered';  ELSIF p\_stage = 4 THEN  lv\_stage\_txt := 'Cancelled';  ELSIF p\_stage = 5 THEN  lv\_stage\_txt := 'Shipped';  END IF;  RETURN lv\_stage\_txt;  END; |
|  |
| CREATE OR REPLACE PROCEDURE status\_check\_sp  (p\_bask IN NUMBER,  p\_stage OUT NUMBER,  p\_desc OUT VARCHAR2)  IS  BEGIN  SELECT idstage  INTO p\_stage  FROM bb\_basketstatus  WHERE idBasket = p\_bask;  p\_desc := status\_desc\_sf(p\_stage);  END; |
|  |
| SELECT object\_name, status  FROM user\_objects  WHERE object\_name IN  ('STATUS\_CHECK\_SP','STATUS\_DESC\_SF'); |
|  |
| CREATE OR REPLACE FUNCTION status\_desc\_sf  (p\_stage IN NUMBER)  RETURN VARCHAR2  IS  lv\_stage\_txt VARCHAR2(30);  BEGIN  IF p\_stage = 1 THEN  lv\_stage\_txt := 'Order Submitted';  ELSIF p\_stage = 2 THEN  lv\_stage\_txt := 'Accepted, sent to shipping';  ELSIF p\_stage = 3 THEN  lv\_stage\_txt := 'Backordered';  ELSIF p\_stage = 4 THEN  lv\_stage\_txt := 'Cancelled';  ELSIF p\_stage = 5 THEN  lv\_stage\_txt := 'Shipped';  ELSIF p\_stage = 6 THEN  lv\_stage\_txt := 'Credit Card Not Approved';  END IF;  RETURN lv\_stage\_txt;  END; |
|  |
| SELECT object\_name, status  FROM user\_objects  WHERE object\_name IN  ('STATUS\_CHECK\_SP','STATUS\_DESC\_SF'); |
|  |
| DECLARE  lv\_stage\_num NUMBER(2);  lv\_desc\_txt VARCHAR2(30);  BEGIN status\_check\_sp(13,lv\_stage\_num,lv\_desc\_txt);  END; |
|  |
| SELECT object\_name, status  FROM user\_objects  WHERE object\_name IN  ('STATUS\_CHECK\_SP','STATUS\_DESC\_SF'); |
|  |
| **Assignment 8-3: Testing Dependencies on Packaged Program Units** |
| CREATE OR REPLACE PACKAGE lookup\_pkg  IS  FUNCTION status\_desc\_pf  (p\_stage IN NUMBER)  RETURN VARCHAR2;  END; |
|  |
| CREATE OR REPLACE PACKAGE BODY lookup\_pkg  IS  FUNCTION status\_desc\_pf  (p\_stage IN NUMBER)  RETURN VARCHAR2  IS  lv\_stage\_txt VARCHAR2(30);  BEGIN  IF p\_stage = 1 THEN  lv\_stage\_txt := 'Order Submitted';  ELSIF p\_stage = 2 THEN  lv\_stage\_txt := 'Accepted, sent to shipping';  ELSIF p\_stage = 3 THEN  lv\_stage\_txt := 'Backordered';  ELSIF p\_stage = 4 THEN  lv\_stage\_txt := 'Cancelled';  ELSIF p\_stage = 5 THEN  lv\_stage\_txt := 'Shipped';  END IF;  RETURN lv\_stage\_txt;  END;  END; |
|  |
| CREATE OR REPLACE PROCEDURE status\_check\_sp  (p\_bask IN NUMBER,  p\_stage OUT NUMBER,  p\_desc OUT VARCHAR2)  IS  BEGIN  SELECT idstage  INTO p\_stage  FROM bb\_basketstatus  WHERE idBasket = p\_bask;  p\_desc := lookup\_pkg.status\_desc\_pf(p\_stage);  END |
|  |
| SELECT status  FROM user\_objects  WHERE object\_name = 'STATUS\_CHECK\_SP'; |
|  |
| The modification did not affect the status of the procedure  Verify that is still VALID |
| CREATE OR REPLACE PACKAGE BODY lookup\_pkg  IS  FUNCTION status\_desc\_pf  (p\_stage IN NUMBER)  RETURN VARCHAR2  IS  lv\_stage\_txt VARCHAR2(30);  BEGIN  IF p\_stage = 1 THEN  lv\_stage\_txt := 'Order Submitted';  ELSIF p\_stage = 2 THEN  lv\_stage\_txt := 'Accepted, sent to shipping';  ELSIF p\_stage = 3 THEN  lv\_stage\_txt := 'Backordered';  ELSIF p\_stage = 4 THEN  lv\_stage\_txt := 'Cancelled';  ELSIF p\_stage = 5 THEN  lv\_stage\_txt := 'Shipped';  ELSIF p\_stage = 6 THEN  lv\_stage\_txt := 'Credit Card Not Approved';  END IF;  RETURN lv\_stage\_txt;  END;  END; |
|  |
| SELECT status  FROM user\_objects  WHERE object\_name = 'STATUS\_CHECK\_SP'; |
|  |
| **Assignment 8-4: Testing Remote Object Dependencies** |
| CREATE DATABASE Link dblink2  USING 'XE' ; |
|  |
| CREATE OR REPLACE FUNCTION status\_desc\_sf  (p\_stage NUMBER)  RETURN VARCHAR2  IS  lv\_stage\_txt VARCHAR2(30);  BEGIN  IF p\_stage = 1 THEN  lv\_stage\_txt := 'Order Submitted';  ELSIF p\_stage = 2 THEN  lv\_stage\_txt := 'Accepted, sent to shipping';  ELSIF p\_stage = 3 THEN  lv\_stage\_txt := 'Backordered';  ELSIF p\_stage = 4 THEN  lv\_stage\_txt := 'Cancelled';  ELSIF p\_stage = 5 THEN  lv\_stage\_txt := 'Shipped';  END IF;  RETURN lv\_stage\_txt;  END; |
|  |
| CREATE OR REPLACE PROCEDURE status\_check\_sp  (p\_bask IN NUMBER,  p\_stage OUT NUMBER,  p\_desc OUT VARCHAR2)  IS  BEGIN  SELECT idstage  INTO p\_stage  FROM bb\_basketstatus  WHERE idBasket = p\_bask;  p\_desc := status\_desc\_sf@dblink2(p\_stage);  END; |
|  |
| SELECT object\_name, status  FROM user\_objects  WHERE  object\_name IN  ('STATUS\_CHECK\_SP') |
|  |
| CREATE OR REPLACE FUNCTION status\_desc\_sf  (p\_stage NUMBER)  RETURN VARCHAR2  IS  lv\_stage\_txt VARCHAR2(30);  BEGIN  IF p\_stage = 1 THEN  lv\_stage\_txt := 'Order Submitted';  ELSIF p\_stage = 2 THEN  lv\_stage\_txt := 'Accepted, sent to shipping';  ELSIF p\_stage = 3 THEN  lv\_stage\_txt := 'Backordered';  ELSIF p\_stage = 4 THEN  lv\_stage\_txt := 'Cancelled';  ELSIF p\_stage = 5 THEN  lv\_stage\_txt := 'Shipped';  ELSIF p\_stage = 6 THEN  lv\_stage\_txt := 'Credit Card Not Approved';  END IF;  RETURN lv\_stage\_txt;  END; |
|  |
| CREATE OR REPLACE PROCEDURE status\_check\_sp  (p\_bask IN NUMBER,  p\_stage OUT NUMBER,  p\_desc OUT VARCHAR2)  IS  BEGIN  SELECT idstage  INTO p\_stage  FROM bb\_basketstatus  WHERE idBasket = p\_bask;  p\_desc := status\_desc\_sf@dblink2(p\_stage);  END; |
|  |
| **Assignment 8-5: Identifying Dependencies** |
| SELECT name, dependency\_type  FROM USER\_DEPENDENCIES |
|  |
| SELECT object\_name, status  FROM USER\_OBJECTS  WHERE object\_type = 'PROCEDURE'; |
|  |
| **Assignment 8-6: Reviewing the utldtree.sql Script** |
| * object\_id number * referenced\_object\_id number * nest\_level number * seq# number |
| **Assignment 8-7: Avoiding Recompilation Errors** |
| * By using the TYPE% and ROWTYPE it instructs the system to apply the datatypes based from the column type at run time. When changing the datatype from NUMBER to a VARCHAR2 remember that it will cause an error because the program code uses a numeric function. * Use the SELECT \* instead of naming each column because if someone makes any changes for example dropping a column, then the query won’t fail if it is SELECT \*; though it does take more processing time. This helps by not having to restructure the code over again. * It is recommended to use names of each column when using an INSERT statement to make changes to a table so that it doesn’t raise an error. If you don’t use a column list, the INSERT value to be entered for each column, which then will raise an error for any table column deletions or additions. |
| **Assignment 8-8: Defining Types of Dependencies** |
| * **Direct Dependency** is a relationship between one database object that references another object for example when a procedure calls to a function * **Indirect dependency** is when the relationship is not immediately connected, example a procedure calls a procedure which calls a function. The first procedure is indirectly depended on the function * **Remote dependencies** are used to connect to other Oracle databases by using a link to call the objects in the other database. These objects are remote objects. When remote objects are modified the local dependent objects don’t show any error as invalid because remote dependencies are not checked until run time. Data dictionary doesn’t track remote dependencies |