

5Laboratory 05

Jeffrey Daniel Leiva Cascante

Tamara Nicole Rodríguez Luna

Computer engineering department, Tecnológico de Costa Rica

IC4301 – Databases I

Teacher: Msc. Adriana Álvarez Figueroa.

September 30th 2022.

In this document are the evidence of laboratory 05 where you can see the objects created, data inserted, modified, among others.

Evidence:

1. Create a procedure (using OPEN of the cursor) that returns all the phone numbers of a person. The person number must be sent as parameter as well as the phone type and the latter can be null, in which In this case, all the telephones of the person must be displayed. 15pts

TABLE PERSON

The screenshot shows the Oracle SQL Developer interface with the PERSON table selected in the central workspace. The table has columns: ID_PERSON, FIRST_NAME, SECOND_NAME, FIRST_SURNAME, and SECOND_SURNAME. The data is as follows:

	ID_PERSON	FIRST_NAME	SECOND_NAME	FIRST_SURNAME	SECOND_SURNAME
1	0	Alberto	Edwin	Morales	Jara
2	1	Ariela	Jimena	Loria	Valverde
3	2	Josue	Andres	Corrales	Villalobos
4	3	Esteban	Francisco	Mora	Rojas
5	4	Leidy	Liliana	Cascante	Solano

TABLE PHONE

The screenshot shows the Oracle SQL Developer interface with the PHONE table selected in the central workspace. The table has columns: ID_PHONE, ID_CATEGORY, ID_PERSON, and PHONE_NUMBER. The data is as follows:

	ID_PHONE	ID_CATEGORY	ID_PERSON	PHONE_NUMBER
1	1	1	1	64854789
2	2	0	0	22707711
3	3	0	1	22145391
4	4	1	0	62872191

TABLE PHONECATEGORY

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows a connection to 'BASEUNO'. In the center, the 'PHONECATEGORY' table is selected in the 'Data' tab, displaying two rows of data:

ID_CATEGORY	CATEGORY_DESCRIPTION
1	0 CASA
2	1 CELULAR

Compilation of getPhones procedure using Open:

The screenshot shows the Oracle SQL Developer interface with the 'procedimientoOpen.sql' script open in the SQL Worksheet. The code defines a procedure 'getPhones' that joins three tables: person, phone, and phonecategory. It uses a cursor to select names, phone numbers, and category descriptions, and then fetches the results into a result set. The procedure is successfully compiled.

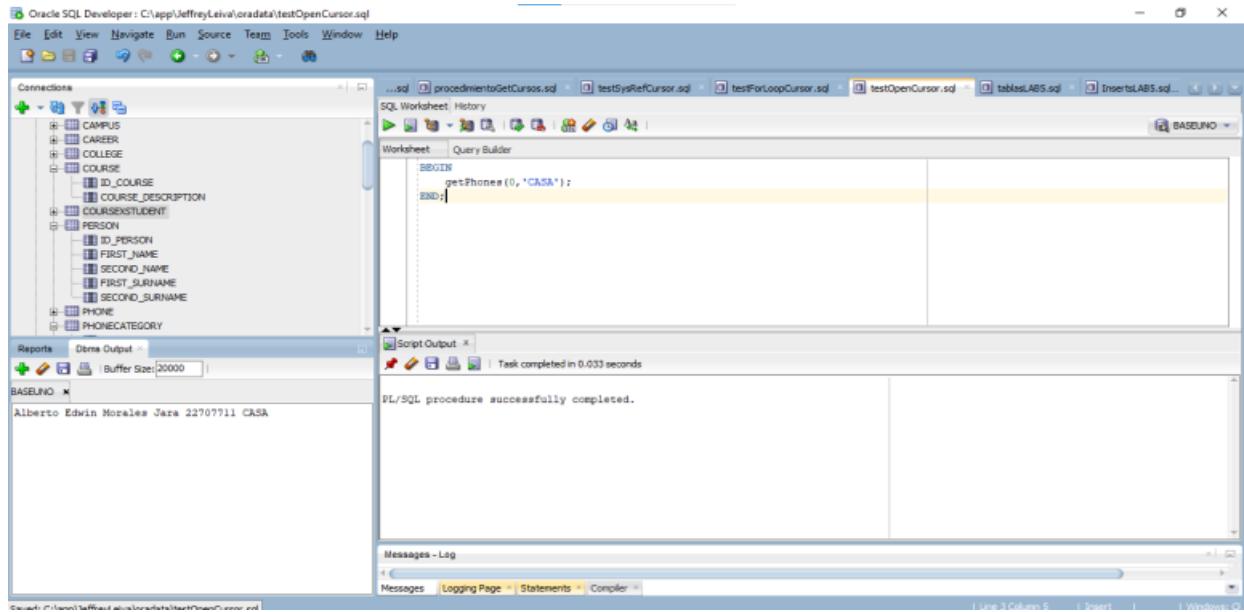
```

CREATE OR REPLACE PROCEDURE getPhones(pnIdPerson IN NUMBER, pvPhoneType IN VARCHAR2)
AS
    vcName VARCHAR2(1000);
    vPhoneNumber NUMBER;
    vcCategoryDescription NUMBER;
    CURSOR phonesCursor IS
        SELECT first_name||' '||second_name||' '||first_surname||' '||second_surname AS name, phone_number, category_description
        FROM person
        INNER JOIN phone
        ON person.id_person = phone.id_person
        INNER JOIN phonecategory
        ON phone.id_category = phonecategory.id_category
        WHERE person.id_person = NVL(pnIdPerson, person.id_person) AND phonecategory.category_description = NVL(pvPhoneType, phonecategory.category_description);
BEGIN
    OPEN phonesCursor;
    LOOP
        FETCH phonesCursor INTO vcName, vPhoneNumber, vcCategoryDescription;
        EXIT WHEN phonesCursor%NOTFOUND;
    END LOOP;
    CLOSE phonesCursor;
END;
  
```

Procedure GETPHONES compiled

2. Create a test to test the procedure using cursors. 20pts

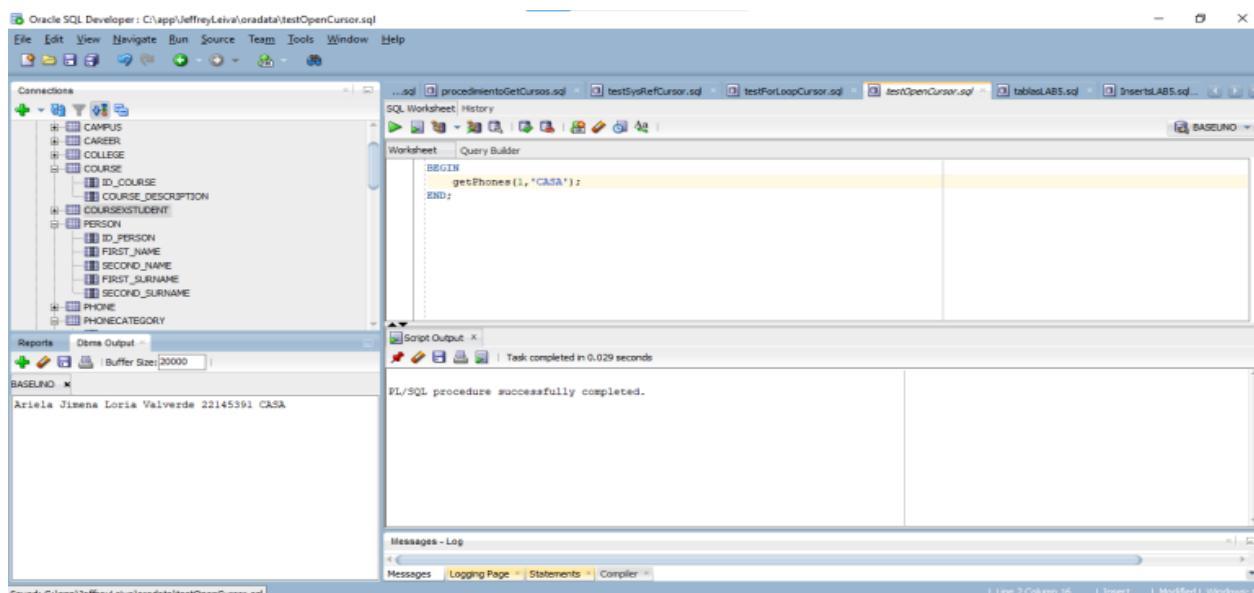
Next there are the test cases for the previous procedure:



The screenshot shows the Oracle SQL Developer interface. The connections pane lists several tables: CAMPUS, CAREER, COLLEGE, COURSE, COURSESTUDENT, PERSON, PHONE, and PHONECATEGORY. The current connection is set to 'BASEUNO'. In the SQL Worksheet, the following PL/SQL code is executed:

```
BEGIN
    getPhones(0, 'CASA');
END;
```

The script output window shows the message: "PL/SQL procedure successfully completed." Below the worksheet, the messages log indicates: "Task completed in 0.033 seconds".



The screenshot shows the Oracle SQL Developer interface with the same connection setup as the first screenshot. The current connection is 'BASEUNO'. In the SQL Worksheet, the following PL/SQL code is executed:

```
BEGIN
    getPhones(1, 'CASA');
END;
```

The script output window shows the message: "PL/SQL procedure successfully completed." Below the worksheet, the messages log indicates: "Task completed in 0.029 seconds".

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\testOpenCursor.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

...sql procedimientoGetCursos.sql testSysRefCursor.sql testForLoopCursor.sql testOpenCursor.sql tablesLAB5.sql InsertsLAB5.sql

SQL Worksheet History

Worksheet Query Builder

```
BEGIN
    getPhones(0, 'CELULAR');
END;
```

Script Output x

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

Saved: C:\app\JeffreyLeiva\pradata\testOpenCursor.sql

Line 2 Column 25 Insert Modified Windows

This screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying a PL/SQL block that calls a procedure named 'getPhones' with parameters (0, 'CELULAR'). The 'Script Output' pane shows the message 'PL/SQL procedure successfully completed.' Below the main workspace are tabs for 'Messages', 'Logging Page', 'Statements', and 'Compiler'. The status bar at the bottom indicates the current line (Line 2) and column (Column 25). The 'Connections' sidebar on the left lists various database objects like CAMPUS, CAREER, COLLEGE, COURSE, PERSON, PHONE, and PHONECATEGORY.

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\testOpenCursor.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

...sql procedimientoGetCursos.sql testSysRefCursor.sql testForLoopCursor.sql testOpenCursor.sql tablesLAB5.sql InsertsLAB5.sql

SQL Worksheet History

Worksheet Query Builder

```
BEGIN
    getPhones(1, 'CELULAR');
END;
```

Script Output x

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

Saved: C:\app\JeffreyLeiva\pradata\testOpenCursor.sql

Line 2 Column 25 Insert Modified Windows

This screenshot shows the Oracle SQL Developer interface, similar to the one above. It displays a slightly different PL/SQL block where the parameter value has been changed from 0 to 1. The 'Script Output' pane again shows the message 'PL/SQL procedure successfully completed.' The rest of the interface, including the connections tree and status bar, is identical to the first screenshot.

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\testOpenCursor.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

...sql procedimientosGetCursos.sql testSysRefCursor.sql testForLoopCursor.sql testOpenCursor.sql tablesLAB5.sql InsertSLABS.sql

SQL Worksheet History

Worksheet Query Builder

```
BEGIN
    getPhones(1,NULL);
END;
```

Script Output X | Task completed in 0.024 seconds

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

Alberto Edwin Morales Jara 22707711 CASA
Alberto Edwin Morales Jara 62872181 CELULAR

BASEUNO x

Connections

...sql procedimientosGetCursos.sql testSysRefCursor.sql testForLoopCursor.sql testOpenCursor.sql tablesLAB5.sql InsertSLABS.sql

SQL Worksheet History

Worksheet Query Builder

```
BEGIN
    getPhones(1,NULL);
END;
```

Script Output X | Task completed in 0.028 seconds

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

Ariela Jimena Loria Valverde 64054789 CELULAR
Ariela Jimena Loria Valverde 22145391 CASA

BASEUNO x

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\testOpenCursor.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

...sql procedimientosGetCursos.sql testSysRefCursor.sql testForLoopCursor.sql testOpenCursor.sql tablesLAB5.sql InsertSLABS.sql

SQL Worksheet History

Worksheet Query Builder

```
BEGIN
    getPhones(1,NULL);
END;
```

Script Output X | Task completed in 0.028 seconds

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

Ariela Jimena Loria Valverde 64054789 CELULAR
Ariela Jimena Loria Valverde 22145391 CASA

BASEUNO x

3. What happens when more fields are added in the test fetch? 10pts

The database throws an exception saying that there is a wrong number of values in the INTO list of Fetch Statement.

4. What happens when the fetch of the test failed to obtain fields? 10pts

The database throws an exception saying that there is a wrong number of values in the INTO list of Fetch Statement.

5. What happens when in the test fetch the fields have a different domain from the what is getting the cursor? 10pts

The procedure compiles and does not throw any exceptions but when you call it from the test query it throws an exception that says there are numeric or value errors.

6. Modify the cursor to use the code with LOOP. 10pts

Compilation of the getPhones Procedure using for loop:

The screenshot shows the Oracle SQL Developer interface with the following details:

- File Bar:** Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\forLoopGetPhones.sql
- Connections:** BASEUNO
- Tables (Filtered):** CAMPUS, CAREER, COLLEGE, COURSE, ID_COURSE, COURSE_DESCRIF, COURSESTUDENT, PERSON, PHONE, PHONECATEGORY, STATUS
- SQL Worksheet:** History
- Code:**

```
CREATE OR REPLACE PROCEDURE getPhonesForLoop(pnIdPerson IN NUMBER, pvPhoneType IN VARCHAR2)
AS
  CURSOR phones (pnIdStudent IN NUMBER, pvPhoneType IN VARCHAR2)
  IS
    SELECT first_name||' '||second_name||' '||first_surname||' '||second_surname AS name, phone_number, category_description
    FROM person
    INNER JOIN phone
    ON person.id_person = phone.id_person
    INNER JOIN phonecategory
    ON phone.id_category = phonecategory.id_category
    WHERE person.id_person = NVL(pnIdPerson, person.id_person) AND phonecategory.category_description = NVL(pvPhoneType, phonecategory.category_description);

BEGIN
  FOR i IN phones(pnIdPerson, pvPhoneType) LOOP
    dbms_output.put_line(i.name||' '||i.phone_number||' '||i.category_description);
  END LOOP;
END getPhonesForLoop;
```
- Script Output:** Procedure GETPHONESFORLOOP compiled (Task completed in 0.075 seconds)
- Messages - Log:**
- Bottom Bar:** Lines 12 Column 11 | Insert | Windows | Help

Next there are various test cases for this point.

The image shows two separate sessions in Oracle SQL Developer. Both sessions are connected to the same database, 'BASEUNO'. The left session shows a query builder window with the following PL/SQL code:

```
BEGIN
    getPhonesForLoop(0, "CELULAR");
END;
```

The right session shows a similar query builder window with the same code. Both sessions have a 'Script Output' tab at the bottom that displays the message 'PL/SQL procedure successfully completed.' This indicates that the procedure executed correctly in both cases.

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\testForLoopCursor.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

...sql procedimientosGetCursos.sql testSysRefCursor.sql testForLoopCursor.sql testOpenCursor.sql tablesLAB5.sql InsertLAB5.sql

Worksheet Query Builder

```
BEGIN
    getPhonesForLoop(0, 'CASA');
END;
```

Script Output x

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

C:\app\JeffreyLeiva\oradata\testForLoopCursor.sql

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\testForLoopCursor.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

...sql procedimientosGetCursos.sql testSysRefCursor.sql testForLoopCursor.sql testOpenCursor.sql tablesLAB5.sql InsertLAB5.sql

Worksheet Query Builder

```
BEGIN
    getPhonesForLoop(1, 'CASA');
END;
```

Script Output x

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

C:\app\JeffreyLeiva\oradata\testForLoopCursor.sql

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\testForLoopCursor.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

...sql procedimientosGetCursos.sql testSysRefCursor.sql testForLoopCursor.sql testOpenCursor.sql tablesLAB5.sql InsertLAB5.sql

Worksheet Query Builder

```
BEGIN
    getPhonesForLoop(1, 'CASA');
END;
```

Script Output x

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\testForLoopCursor.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

...ad procedimientosGetCursos.sql testSysRefCursor.sql testForLoopCursor.sql testOpenCursor.sql tablasLAB5.sql InsertaLAB5.sql

SQL Worksheet: History

Worksheet Query Builder

```
BEGIN
    getPhonesForLoop(0,NULL);
END;
```

Script Output x

Alberto Edwin Morales Jara 22707711 CASA
Alberto Edwin Morales Jara 62872191 CELULAR

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\testForLoopCursor.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

...ad forLoopGetPhones.sql procedimientosGetCursos.sql testSysRefCursor.sql testForLoopCursor.sql testOpenCursor.sql tablasLAB5.sql InsertaLAB5.sql

SQL Worksheet: History

Worksheet Query Builder

```
BEGIN
    getPhonesForLoop(1,NULL);
END;
```

Script Output x

Ariela Jimena Loria Valverde 64654799 CELULAR
Ariela Jimena Loria Valverde 22145391 CASA

PL/SQL procedure successfully completed.

Messages - Log

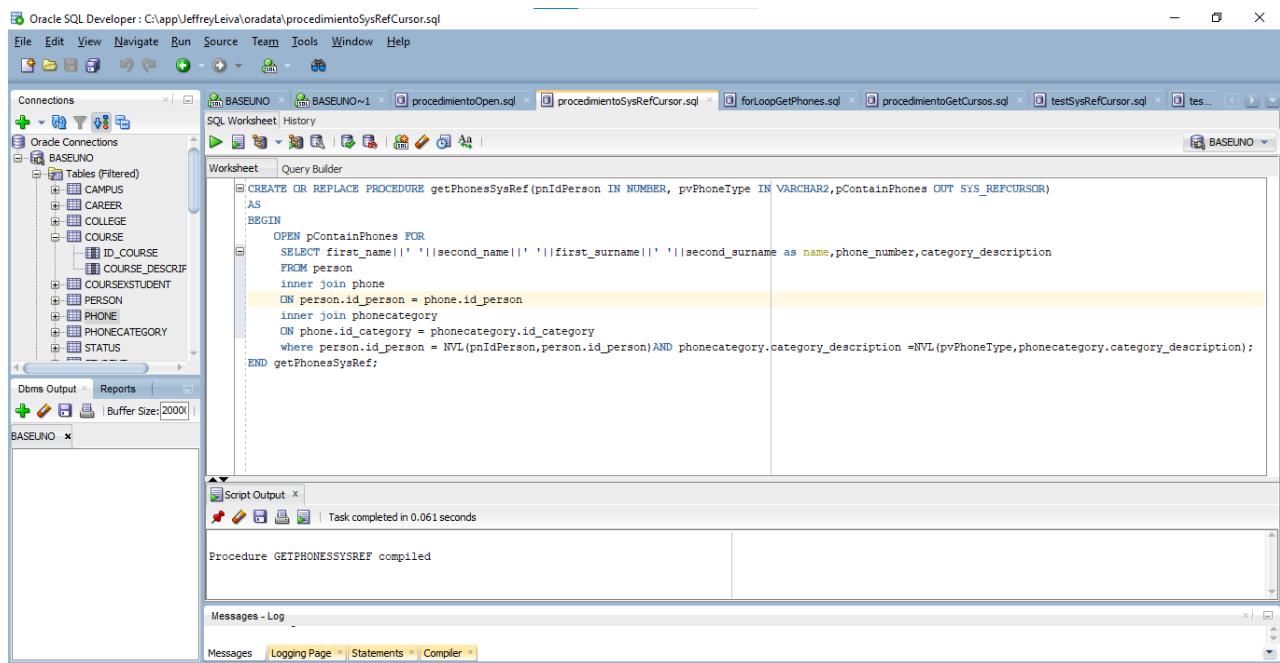
Messages Logging Page Statements Compiler

Saved: C:\app\JeffreyLeiva\oradata\testForLoopCursor.sql

Line 2 Column 28 Insert Windows: C

7. Modify the cursor to use the code from SYS_REF_CURSOR.10pts

Compilation for the getPhones procedure using sys_refcursor:

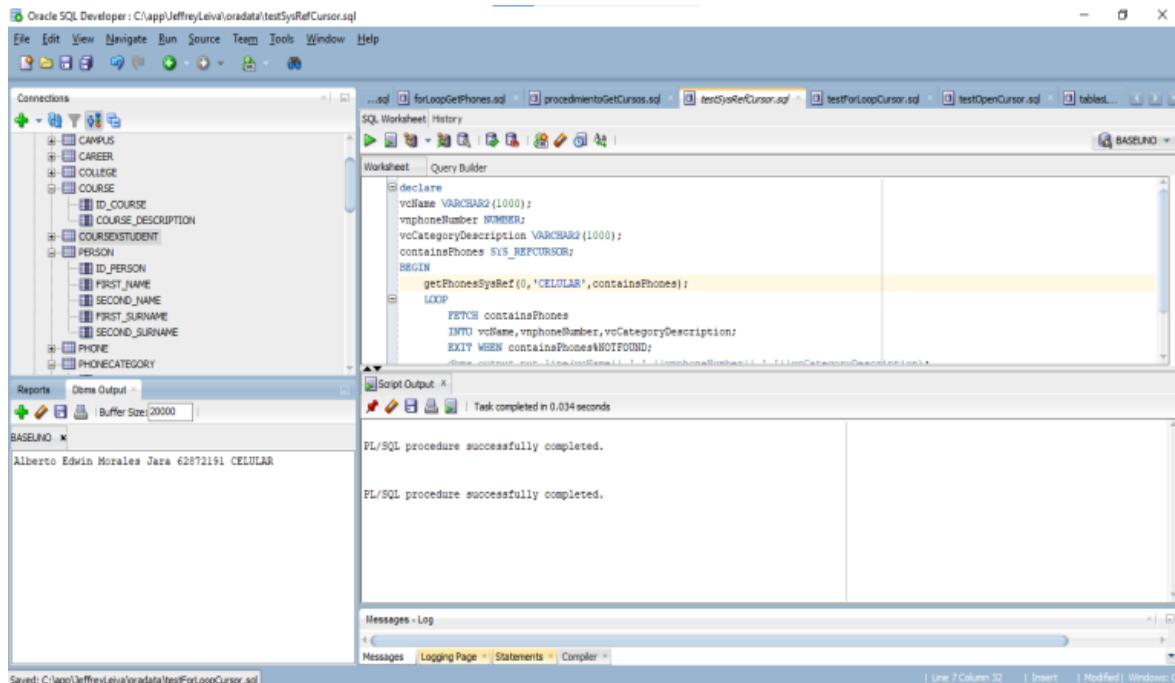


The screenshot shows the Oracle SQL Developer interface with the following details:

- File Edit View Navigate Run Source Team Tools Window Help**
- Connections** pane: Oracle Connections, BASEUNO, Tables (Filtered) including CAMPUS, CAREER, COLLEGE, COURSE, COURSESTUDENT, PERSON, PHONE, PHONECATEGORY, STATUS.
- Worksheet**: The code for the procedure is pasted here:

```
CREATE OR REPLACE PROCEDURE getPhonesSysRef(pnIdPerson IN NUMBER, pvPhoneType IN VARCHAR2,pContainPhones OUT SYS_REFCURSOR)
AS
BEGIN
  OPEN pContainPhones FOR
    SELECT first_name||'('||second_name||')'||first_surname||' ('||second_surname as name,phone_number,category_description
    FROM person
    inner join phone
    ON person.id_person = phone.id_person
    inner join phonecategory
    ON phone.id_category = phonecategory.id_category
    where person.id_person = NVL(pnIdPerson,person.id_person)AND phonecategory.category_description =NVL(pvPhoneType,phonecategory.category_description);
END getPhonesSysRef;
```
- Script Output**: Task completed in 0.061 seconds.
- Messages - Log**: Procedure GETPHONESYSREF compiled.
- Messages**, **Logging Page**, **Statements**, **Compiler** tabs.

Next there are some test cases



The screenshot shows the Oracle SQL Developer interface with the following details:

- File Edit View Navigate Run Source Team Tools Window Help**
- Connections** pane: Oracle Connections, BASEUNO, Tables (Filtered) including CAMPUS, CAREER, COLLEGE, COURSE, COURSESTUDENT, PERSON, PHONE, PHONECATEGORY.
- Worksheet**: The test code is pasted here:

```
declare
  vName VARCHAR2(1000);
  vPhoneNumber NUMBER;
  vcCategoryDescription VARCHAR2(1000);
  containsPhones SYS_REFCURSOR;
BEGIN
  getPhonesSysRef(0,'CELULAR',containsPhones);
  LOOP
    FETCH containsPhones
    INTO vName,vPhoneNumber,vcCategoryDescription;
    EXIT WHEN containsPhones%NOTFOUND;
  END LOOP;
END;
```
- Script Output**: Task completed in 0.034 seconds.
- Messages - Log**: PL/SQL procedure successfully completed.
- Messages**, **Logging Page**, **Statements**, **Compiler** tabs.

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\testSysRefCursor.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

...sql forLoopGetPhones.sql procedimientosGetCursos.sql testSysRefCursor.sql testForLoopCursor.sql testOpenCursor.sql tablas...

SQL Worksheet History

Worksheet Query Builder

```
declare
  vcName VARCHAR2(1000);
  vPhoneNumber NUMBER;
  vcCategoryDescription VARCHAR2(1000);
  containsPhones SYS_REFCURSOR;
BEGIN
  getPhonesSysRef('CELULAR',containsPhones);
  LOOP
    FETCH containsPhones
    INTO vcName,vPhoneNumber,vcCategoryDescription;
    EXIT WHEN containsPhones%NOTFOUND;
    dbms_output.put_line(vcName||'-'||vPhoneNumber||'-'||vcCategoryDescription);
END;
```

Script Output

PL/SQL procedure successfully completed.

Messages - Log

Saved: C:\app\JeffreyLeiva\oradata\testForLoopCursor.sql

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\testSysRefCursor.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

...sql forLoopGetPhones.sql procedimientosGetCursos.sql testSysRefCursor.sql testForLoopCursor.sql testOpenCursor.sql tablas...

SQL Worksheet History

Worksheet Query Builder

```
declare
  vcName VARCHAR2(1000);
  vPhoneNumber NUMBER;
  vcCategoryDescription VARCHAR2(1000);
  containsPhones SYS_REFCURSOR;
BEGIN
  getPhonesSysRef('CASA',containsPhones);
  LOOP
    FETCH containsPhones
    INTO vcName,vPhoneNumber,vcCategoryDescription;
    EXIT WHEN containsPhones%NOTFOUND;
    dbms_output.put_line(vcName||'-'||vPhoneNumber||'-'||vcCategoryDescription);
END;
```

Script Output

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

Line 3 Column 14 | Insert | Modified | Windows | Close

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\testSysRefCursor.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

- CAMPUS
- CAREER
- COLLEGE
- COURSE
 - ID_COURSE
 - COURSE_DESCRIPTION
- COURSESTUDENT
- PERSON
 - ID_PERSON
 - FIRST_NAME
 - SECOND_NAME
 - FIRST_SURNAME
 - SECOND_SURNAME
- PHONE
- PHONECATEGORY

Reports Dmbs Output

BASEUNO * Ariela Jimena Loria Valverde 22145391 CASA

SQL Worksheet History

Worksheet Query Builder

```
declare
    vcName VARCHAR2(1000);
    vPhoneNumber NUMBER;
    vcCategoryDescription VARCHAR2(1000);
    containsPhones SYS_REFCURSOR;
BEGIN
    getPhonesSysRef(1,'CASA',containsPhones);
    LOOP
        FETCH containsPhones
        INTO vcName,vPhoneNumber,vcCategoryDescription;
        EXIT WHEN containsPhones%NOTFOUND;
        dbms_output.put_line(vcName||'-'||vPhoneNumber||'-'||vcCategoryDescription);
    END LOOP;
END;
```

Script Output x

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\testSysRefCursor.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

- CAMPUS
- CAREER
- COLLEGE
- COURSE
 - ID_COURSE
 - COURSE_DESCRIPTION
- COURSESTUDENT
- PERSON
 - ID_PERSON
 - FIRST_NAME
 - SECOND_NAME
 - FIRST_SURNAME
 - SECOND_SURNAME
- PHONE
- PHONECATEGORY

Reports Dmbs Output

BASEUNO * Alberto Edwin Morales Jara 22707711 CASA
Alberto Edwin Morales Jara 62872191 CELULAR

SQL Worksheet History

Worksheet Query Builder

```
declare
    vcName VARCHAR2(1000);
    vPhoneNumber NUMBER;
    vcCategoryDescription VARCHAR2(1000);
    containsPhones SYS_REFCURSOR;
BEGIN
    getPhonesSysRef(0,NULL,containsPhones);
    LOOP
        FETCH containsPhones
        INTO vcName,vPhoneNumber,vcCategoryDescription;
        EXIT WHEN containsPhones%NOTFOUND;
        dbms_output.put_line(vcName||'-'||vPhoneNumber||'-'||vcCategoryDescription);
    END LOOP;
END;
```

Script Output x

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

```

declare
    veName VARCHAR2(1000);
    vphoneNumber NUMBER;
    vcCategoryDescription VARCHAR2(1000);
    containsPhones SYS_REFCURSOR;
BEGIN
    getPhonesSysRef(i, NULL, containsPhones);
    LOOP
        FETCH containsPhones
        INTO vcName, vphoneNumber, vcCategoryDescription;
        EXIT WHEN containsPhones%NOTFOUND;
    END LOOP;
END;

```

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar lists several tables under the 'BASEUNO' schema, including CAMPUS, CAREER, COLLEGE, COURSE, PERSON, PHONE, and PHONECATEGORY. The 'Worksheet' tab displays the PL/SQL code above. The 'Script Output' tab shows the message 'PL/SQL procedure successfully completed.' The 'Messages - Log' tab is also visible.

8. Create the UN schema and create the student, courses and their associations tables. Create a cursor getcursos that returns the courses of a student from the student_id and the status of the course (either approved, failed, pending) where this last parameter can be null in which case it must return all the student's courses and the status of it. 15pt

Creation of UN tablespace:

```

CREATE TABLESPACE UN_DATA
DATAFILE 'C:\app\JeffreyLeiva\oradata\demo\undata.dbf'
SIZE 10M
REUSE
AUTOEXTEND ON
NEXT 512K
MAXSIZE 200M;

CREATE TABLESPACE UN_Ind
DATAFILE 'C:\app\JeffreyLeiva\oradata\demo\unInd.dbf'
SIZE 10M
REUSE
AUTOEXTEND ON
NEXT 512K
MAXSIZE 200M;

```

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar lists tables under the 'BASEUNO' schema. The 'Worksheet' tab contains the SQL code for creating two tablespaces, 'UN_DATA' and 'UN_Ind'. The 'Script Output' tab is empty. The 'Messages - Log' tab is visible at the bottom.

Creation of UN schema

The screenshot shows the Oracle SQL Developer interface. In the top menu bar, the path is C:\app\JeffreyLeiva\oradata\createUNSchema.sql. The Connections pane shows a connection to BASEUNO. The Worksheet pane contains the following SQL code:

```
CREATE USER un
IDENTIFIED BY un
DEFAULT TABLESPACE un_data
QUOTA 10M ON un_data
TEMPORARY TABLESPACE temp
QUOTA 5M ON system;

-- ROLES
GRANT "CONNECT" TO "UN" ;

-- SYSTEM PRIVILEGES
GRANT ALTER ANY INDEX TO "UN" ;
GRANT DROP ANY SEQUENCE TO "UN" ;
GRANT ALTER ANY PROCEDURE TO "UN" ;
GRANT CREATE ANY PROCEDURE TO "UN" ;
GRANT CREATE ANY INDEX TO "UN" ;
GRANT ALTER ANY INDEX TO "UN" ;
GRANT CREATE ANY SEQUENCE TO "UN" ;
GRANT CREATE ANY SEQUENCE TO "UN" ;
GRANT CREATE VIEW TO "UN" ;
GRANT ALTER ANY TABLE TO "UN" ;
GRANT CREATE SESSION TO "UN" ;
GRANT SELECT ANY TABLE TO "UN" ;
GRANT DELETE ANY TABLE TO "UN" ;
GRANT ALTER ANY SEQUENCE TO "UN" ;
GRANT CREATE TABLE TO "UN" ;
GRANT DROP ANY TABLE TO "UN" ;
```

The bottom status bar shows the file is saved at C:\app\JeffreyLeiva\oradata\procedimientoOpen.sql.

TABLE COURSE

The screenshot shows the Oracle SQL Developer interface. In the top menu bar, the path is C:\app\JeffreyLeiva\oradata\createUNSchema.sql. The Connections pane shows a connection to BASEUNO. The Worksheet pane shows the following table structure and data:

ID_COURSE	COURSE_DESCRIPTION
1	0 Matemáticas
2	2 Ajedrez
3	3 Ambiente Humano
4	4 Cultural

The bottom status bar shows the file is saved at C:\app\JeffreyLeiva\oradata\procedimientoOpen.sql.

TABLE COURSEXSTUDENT

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema for the 'BASEBONO' connection, with the 'COURSEXSTUDENT' table selected. The right pane shows the data grid for the 'COURSEXSTUDENT' table, which has three columns: ID_COURSE, ID_STUDENT, and ID_STATUS. The data is as follows:

ID_COURSE	ID_STUDENT	ID_STATUS
1	0	0
2	2	0
3	3	1
4	4	1

TABLE STUDENT

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema for the 'BASEBONO' connection, with the 'STUDENT' table selected. The right pane shows the data grid for the 'STUDENT' table, which has five columns: ID_STUDENT, STUDENT_CARD, FIRST_NAME, SECOND_NAME, and FIRST_SURNAME. The data is as follows:

ID_STUDENT	STUDENT_CARD	FIRST_NAME	SECOND_NAME	FIRST_SURNAME
1	0	20210167	Jeffrey	Daniel
2	1	20210921	Tamara	Nicole

TABLE STATUS

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, the connection is set to 'BASEUNO'. The left sidebar displays the database schema, including tables like FIRST_NAME, SECOND_NAME, PHONE, PHONECATEGORY, STATUS, and STUDENT. The central pane shows the 'STATUS' table with three rows:

ID_STATUS	STATUS_DESCRIPTION
1	O APROBADO
2	I REPROBADO
3	P PENDIENTE

The bottom pane shows the 'Messages - Log' and 'Script Output' tabs.

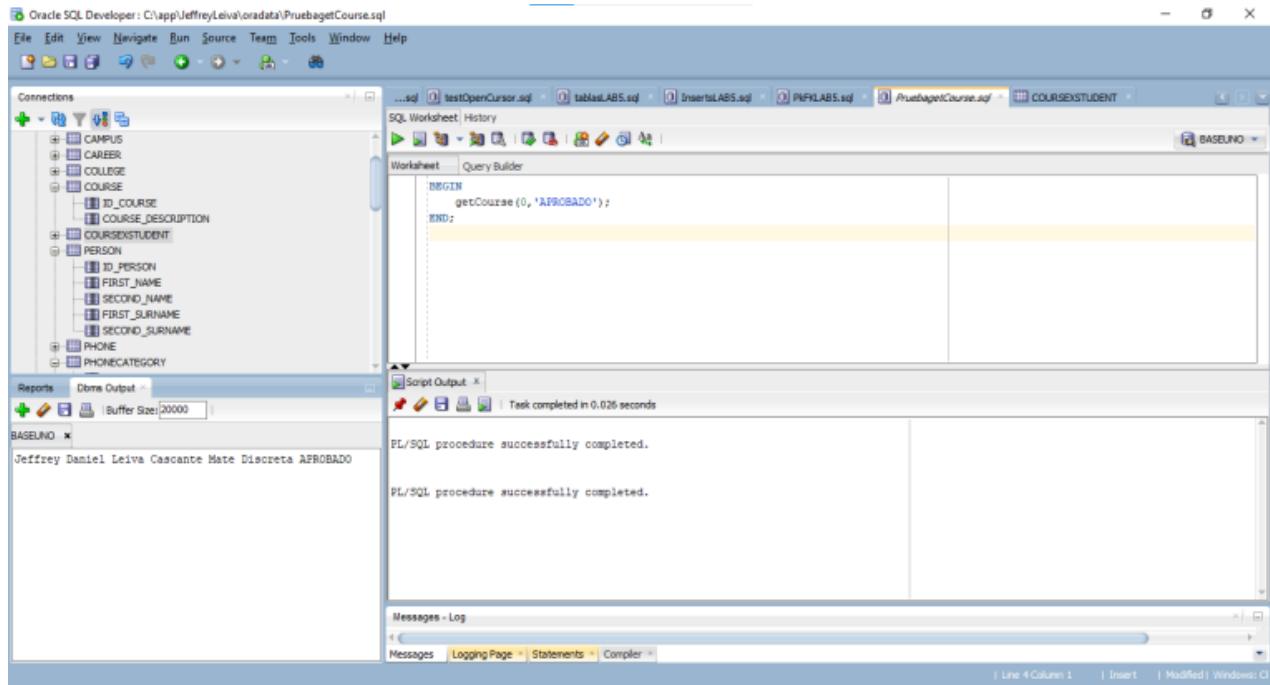
PROCEDURE GETCOURSE

The screenshot shows the Oracle SQL Developer interface with the connection set to 'BASEUNO'. The left sidebar shows the database schema with tables such as CAMPUS, CAREER, COLLEGE, COURSE, PERSON, PHONE, PHONECATEGORY, and STATUS. The central pane displays the SQL code for creating a stored procedure:

```
CREATE OR REPLACE PROCEDURE getCourse(pnIdStudent IN NUMBER, pcStateCourse IN VARCHAR2)
AS
    vcStateCourse VARCHAR2(1000);
    vcCourseName VARCHAR2(1000);
    vclName VARCHAR2(1000);
    CURSOR coursesCursor IS
        SELECT first_name||'||second_name||'||first_surname||'||second_surname AS name, course.course_description, status.status_description
        FROM student
        INNER JOIN coursestudent
        ON student.id_student = coursestudent.id_student
        INNER JOIN course
        ON course.id_course = coursestudent.id_course
        INNER JOIN status
        ON coursestudent.id_status = status.id_status
        WHERE student.id_student = NVL(pnIdStudent, student.id_student) AND status.status_description = NVL(pcStateCourse, status.status_description);
BEGIN
    OPEN coursesCursor;
    LOOP
        FETCH coursesCursor INTO vcName vcCourseName vcStateCourse;
        ...
    END LOOP;
    CLOSE coursesCursor;
END;
```

The 'Script Output' tab shows the message: 'Procedure GETCOURSE compiled'. The bottom pane shows the 'Messages - Log' and 'Script Output' tabs.

Next there are some test cases for the procedure.



Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\PruebagetCourse.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

- CAMPUS
- CAREER
- COLLEGE
- COURSE
 - ID_COURSE
 - COURSE_DESCRIPTION
- COURSESTUDENT
- PERSON
 - ID_PERSON
 - FIRST_NAME
 - SECOND_NAME
 - FIRST_SURNAME
 - SECOND_SURNAME
- PHONE
- PHONECATEGORY

Reports Dbsm Output Buffer Size:20000

BASEUNO Jeffrey Daniel Leiva Cascante Mate Discreta APROBADO

Worksheet Query Builder

```
BEGIN
    getCourse(0, 'APROBADO');
END;
```

Script Output X Task completed in 0.026 seconds

PL/SQL procedure successfully completed.

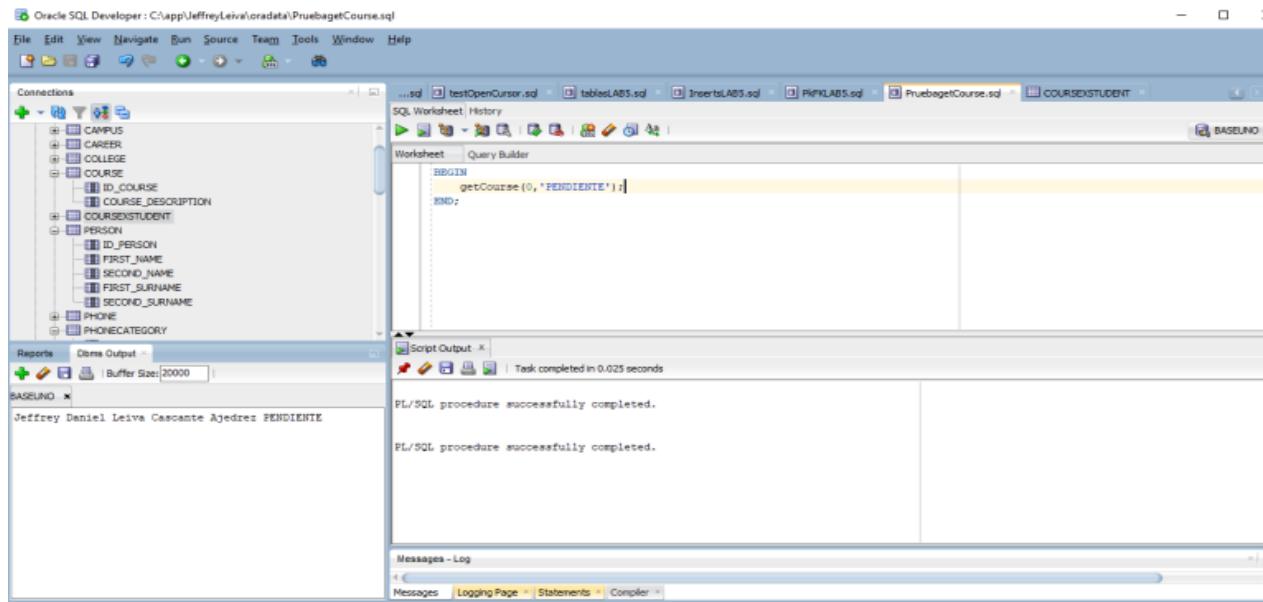
PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

Line 4 Column 1 Insert Modified Windows CI

The screenshot shows the Oracle SQL Developer interface. In the 'Worksheet' tab, a PL/SQL block is run to execute the 'getCourse' procedure with the parameter 'APROBADO'. The output window shows two messages indicating the procedure was successfully completed. The 'Messages - Log' panel at the bottom shows the status of various components.



Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\PruebagetCourse.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

- CAMPUS
- CAREER
- COLLEGE
- COURSE
 - ID_COURSE
 - COURSE_DESCRIPTION
- COURSESTUDENT
- PERSON
 - ID_PERSON
 - FIRST_NAME
 - SECOND_NAME
 - FIRST_SURNAME
 - SECOND_SURNAME
- PHONE
- PHONECATEGORY

Reports Dbsm Output Buffer Size:20000

BASEUNO Jeffrey Daniel Leiva Cascante Ajedres PENDIENTE

Worksheet Query Builder

```
BEGIN
    getCourse(0, 'PENDIENTE');
END;
```

Script Output X Task completed in 0.025 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

The screenshot shows the Oracle SQL Developer interface. In the 'Worksheet' tab, a PL/SQL block is run to execute the 'getCourse' procedure with the parameter 'PENDIENTE'. The output window shows two messages indicating the procedure was successfully completed. The 'Messages - Log' panel at the bottom shows the status of various components.

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\PruebagetCourse.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

...sql testOpenCursor.sql tablesLAB5.sql InsertsLAB5.sql PHPLAB5.sql PruebagetCourse.sql COURSESTUDENT

SQL Worksheet History

Worksheet Query Builder

```
BEGIN
    getCourse(1, 'REPROBADO');
END;
```

Script Output x

Task completed in 0.026 seconds

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

Reports Demo Output Buffer Size: 20000

BASEUNO *

Tamara Nicole Rodriguez Luna Ambiente Humano REPROBADO

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\PruebagetCourse.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

...sql testOpenCursor.sql tablesLAB5.sql InsertsLAB5.sql PHPLAB5.sql PruebagetCourse.sql COURSESTUDENT

SQL Worksheet History

Worksheet Query Builder

```
BEGIN
    getCourse(0, NULL);
END;
```

Script Output x

Task completed in 0.028 seconds

PL/SQL procedure successfully completed.

Messages - Log

Messages Logging Page Statements Compiler

Reports Demo Output Buffer Size: 20000

BASEUNO *

Jeffrey Daniel Leiva Cascante Mate Discreta APROBADO

Jeffrey Daniel Leiva Cascante Ayedres PENDIENTE

Oracle SQL Developer : C:\app\JeffreyLeiva\oradata\PruebagetCourse.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

...sql testOpenCursor.sql tablesLABS.sql InsertsLABS.sql PNLKLABS.sql PruebagetCourse.sql COURSESTUDENT BASEUNO

SQL Worksheet History

Worksheet Query Builder

```
BEGIN
    getCourse(1,NULL);
END;
```

Script Output X

Task completed in 0.025 seconds

PL/SQL procedure successfully completed.

Reports Data Output

BASEUNO x

Tamara Nicole Rodriguez Luna Ambiente Humano REPROBADO
Tamara Nicole Rodriguez Luna Cultural APROBADO

Buffer Size: 20000

Messages - Log

Messages Logging Page Statements Compiler

This screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The Connections sidebar lists various database objects: CAMPUS, CAREER, COLLEGE, COURSE (with sub-items ID_COURSE, COURSE_DESCRIPTION), COURSESTUDENT, PERSON (with sub-items ID_PERSON, FIRST_NAME, SECOND_NAME, FIRST_SURNAME, SECOND_SURNAME), PHONE, and PHONECATEGORY. The central workspace is a SQL Worksheet titled 'History' containing a single PL/SQL block:BEGIN
 getCourse(1,NULL);
END;

```
Below the worksheet is a Script Output window showing the message "Task completed in 0.025 seconds". The bottom status bar indicates "PL/SQL procedure successfully completed.". On the left, there's a Reports section and a Data Output section for a connection named "BASEUNO", which displays two rows of student information: Tamara Nicole Rodriguez Luna with grades REPROBADO and APROBADO respectively. The bottom of the screen features a Messages - Log panel with tabs for Messages, Logging Page, Statements, and Compiler.
```