Using Cursors create a package that includes stored procedures to implement natural join on any 2 tables (Orders and Ordered Itmes for example).

Notes:

1. Do not send a join query to the database

2. send onls select \* form table in each cursor

3, you will need 2 cursors

4. Can be done in groups.

Due 11/16/2014 (no exceptions).

**SELECT \* FROM EMPLOYEES NATURAL JOIN JOBS;**

drop table natural\_join;

CREATE TABLE natural\_join(

EMPLOYEE\_ID NUMBER(6,0),

FIRST\_NAME VARCHAR2(20 BYTE),

LAST\_NAME VARCHAR2(25 BYTE),

JOB\_ID VARCHAR2(10 BYTE),

JOB\_TITLE VARCHAR2(35 BYTE),

MIN\_SALARY NUMBER(6,0),

MAX\_SALARY NUMBER(6,0)

);

-- Create Package

CREATE OR REPLACE PACKAGE pkg\_join

IS

/\* Define the REF CURSOR type. \*/

TYPE join\_cursor IS REF CURSOR;

PROCEDURE sp\_join\_cursor (emp\_cursor OUT join1\_cursor, job\_cursor OUT join2\_cursor);

END pkg\_join;

-- Create Package Body

CREATE OR REPLACE PACKAGE Body pkg\_join

AS

PROCEDURE sp\_join\_cursor (emp\_cursor OUT join1\_cursor, job\_cursor OUT join2\_cursor)

AS

DECLARE

CURSOR emp\_cursor IS

SELECT \* FROM employees;

r\_emp emp\_cursor%ROWTYPE;

CURSOR job\_cursor IS

SELECT \* FROM jobs

WHERE r\_emp.job\_id = job\_id;

r\_job job\_cursor%ROWTYPE;

BEGIN

OPEN emp\_cursor;

LOOP

FETCH emp\_cursor INTO r\_emp;

EXIT WHEN emp\_cursor%NOTFOUND;

IF job\_cursor%ISOPEN THEN

CLOSE emp\_cursor;

END IF;

OPEN job\_cursor;

LOOP

FETCH job\_cursor INTO r\_job;

EXIT WHEN job\_cursor%NOTFOUND;

END LOOP;

CLOSE job\_cursor;

INSERT INTO natural\_join

VALUES(r\_emp.employee\_id, r\_emp.first\_name, r\_emp.last\_name,

r\_job.job\_id, r\_job.job\_title, r\_job.min\_salary, r\_job.max\_salary);

END LOOP;

CLOSE emp\_cursor;

END;

select \* from natural\_join order by EMPLOYEE\_ID;

select employee\_id, first\_name, last\_name, job\_id, job\_title, min\_salary, max\_salary

from employees NATURAL join jobs order by EMPLOYEE\_ID;