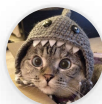


WUOLAH



_david

www.wuolah.com/student/_david



6150

4-PL-SQL.pdf

Ejercicios



3º Administración de Bases de Datos



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

Linguaskill 
from Cambridge

Ya puedes sacarte tu B1/B2/C1 de inglés desde casa

Demuestra tu nivel en 48 horas

Nuevo

#LinguaskillEnCasa

1. Cree una tabla llamada DATOS que sigue el siguiente esquema:

CODIGO	NUMBER(9)
NOMBRE	VARCHAR2(15)
FECHA	DATE

2. Cree una secuencia llamada SECUENCIA_DATOS que comienza en el valor 100 y se incrementa de 3 en 3.

3. Cree una función NUM_ALEATORIO que tiene un argumento N de tipo NUMBER y produce un número aleatorio con N dígitos. Utilizar el paquete DBMS_RANDOM,

4. Cree una función llamada CADENA_ALEATORIA con un argumento N de tipo NUMBER produce una cadena aleatoria de caracteres en mayúscula de longitud N.

5. Cree una función llamada CALCULAR_FECHA que recibe una fecha F, un día de la semana D y un número N y produce la siguiente fecha que cae en D a partir de F incrementada en N días.

6. Cree un procedimiento llamado RELLENAR que recibe como parámetro un número menor que 100 e introduce en DATOS ese número de tuplas. El valor del CODIGO se calcula a partir del valor de la secuencia. El nombre se genera como una cadena aleatoria usando CADENA_ALEATORIA ya creada en un punto anterior de esta relación. Para introducir la fecha use la función CALCULAR_FECHA del apartado anterior a partir de los datos de la fecha actual y para el número N use una llamada a NUM_ALEATORIO. Si el argumento es mayor que 100 se generan solamente 100 tuplas nuevas.

7. Cree una tabla llamada TB_OBJETOS con los siguientes atributos: NOMBRE, CODIGO, FECHA_CREACION, FECHA_MODIFICACION, TIPO, ESQUEMA_ORIGINAL. Recorra la vista ALL_OBJECTS y rellene esta tabla con los datos que se aportan en la vista. Use un cursor y no un INSERT.

8. Cree una tabla TB_ESTILO con los siguientes atributos: TIPO_OBJETO, PREFIJO. En esta tabla se guardan unas normas de estilo de modo que a cada tipo de objeto le corresponde un prefijo en su identificador. Así por ejemplo guardamos la tupla ('PROCEDURE','PR_') para indicar que un nombre correcto de procedimiento es PR_HOLA_MUNDO.

9. Extienda el esquema de la tabla TB_OBJETOS en dos atributos: ESTADO y NOMBRE_CORRECTO. Cree un procedimiento llamado PR_COMPROBAR(ESQUEMA IN VARCHAR2) que recorre la tabla TB_OBJETOS y comprueba si se cumplen las normas de estilo según la tabla TB_ESTILO. El parámetro que recibe es el identificador del esquema sobre el que queremos comprobar las normas. Si no se especifica, se comprueba en todos. Actualice el atributo ESTADO de la tabla TB_OBJETOS con los valores CORRECTO o INCORRECTO según las normas de estilo y el atributo NOMBRE_CORRECTO con el identificador con el prefijo adecuado. El nuevo identificador se calcula anteponiendo el prefijo correcto al identificador antiguo. Si el identificador nuevo excede el tamaño del OBJECT_NAME de Oracle, entonces puede el nuevo identificador por la derecha. Use un cursor de actualización para realizar este procedimiento.

10. Cree una tabla llamada TB_ERRORES con los atributos FECHA, RUTINA, CODIGO y MENSAJE. Cree varios procedimientos que producen errores de Oracle y guarda en la tabla TB_ERRORES un rastro de dichos errores. Así por ejemplo podemos crear el procedimiento:

```
CREATE PROCEDURE PR_SELECT_MAS_UNA_FILA AS
VAR_FILA ALL_TABLES%ROWTYPE;
BEGIN
SELECT * INTO VAR_FILA
FROM ALL_TABLES;
END PR_SELECT_MAS_UNA_FILA;
```

Introduzca en estos procedimientos una sección de excepciones para introducir en TB_ERRORES la aparición de este error:

```
ORA-01422: la recuperación exacta devuelve un número mayor de filas que el solicitado
```

```
/* 1
```

Cree una tabla llamada DATOS que sigue el siguiente esquema:

```
*/
```

```
CREATE USER USUARIO2 IDENTIFIED BY USUARIO2
```

```
QUOTA UNLIMITED ON USERS;
```

```
GRANT CONNECT, CREATE TABLE, CREATE VIEW, CREATE PROCEDURE TO USUARIO2;
```

```
CREATE TABLE USUARIO2.DATOS (
```

```
    CODIGO    NUMBER(9),
```

```
    NOMBRE    VARCHAR2(15),
```

```
    FECHA     DATE
```

```
);
```

```
/
```

```
/* 2
```

Cree una secuencia llamada SECUENCIA_DATOS que comienza en el valor 100 y se incrementa de 3 en 3.

```
*/
```

```
CREATE SEQUENCE USUARIO2.SECUENCIA_DATOS
```

```
START WITH 100
```

```
INCREMENT BY 3;
```

```
/* 3
```

Cree una función NUM_ALEATORIO que tiene un argumento N de tipo NUMBER y produce un número aleatorio con N dígitos. Utilizar el paquete DBMS_RANDOM,

```
*/
```

```
CREATE OR REPLACE FUNCTION USUARIO2.NUM_ALEATORIO (N IN NUMBER)
```

```
RETURN NUMBER IS
```

```
BEGIN
```

```
RETURN TRUNC (DBMS_RANDOM.VALUE (POWER(10, N - 1), POWER(10, N) - 1));
```

```
END NUM_ALEATORIO;
```

```
/*
```

```
SELECT NUM_ALEATORIO(2) FROM DUAL;
```

```
*/
```

```
/* 4
```

Cree una función llamada CADENA_ALEATORIA con un argumento N de tipo NUMBER produce una cadena aleatoria de caracteres en mayúscula de longitud N.

```
*/
```

```
CREATE OR REPLACE FUNCTION USUARIO2.CADENA_ALEATORIA (N IN NUMBER)
```

```
RETURN VARCHAR2 IS
```

```
BEGIN
```

```
RETURN DBMS_RANDOM.STRING('U', N);
```

```
END CADENA_ALEATORIA;
```

```
/*
```

```
SELECT CADENA_ALEATORIA(2) FROM DUAL;
```

```
*/
```

```
/* 5
```

Cree una función llamada CALCULAR_FECHA que recibe una fecha F, un día de la semana D y un número N y produce la siguiente fecha que cae en D a partir de F incrementada en N días.

```
*/
```

```
CREATE OR REPLACE FUNCTION USUARIO2.CALCULAR_FECHA (F IN DATE, D IN VARCHAR2, N IN NUMBER)
```

```
RETURN DATE
```

```
IS
```

```
N_DATE DATE := F + N;
```

```
BEGIN
```

```
RETURN NEXT_DAY(F, D) + N;
```

```
END CALCULAR_FECHA;
```

```
/*
```

```
SELECT CALCULAR_FECHA('27/04/2017', 'VIERNES', 5) FROM DUAL;
```

```
*/
```

```
/* 6
```

Cree un procedimiento llamado RELLENAR que recibe como parámetro un número menor que 100 e introduce en DATOS ese número de tuplas.

El valor del CODIGO se calcula a partir del valor de la secuencia.

El nombre se genera como una cadena aleatoria usando CADENA_ALEATORIA ya creada en un punto anterior de esta relación.

Para introducir la fecha use la función CALCULAR_FECHA del apartado anterior a partir de los datos de la fecha actual y



para el número N use una llamada a NUM_ALEATORIO.

Si el argumento es mayor que 100 se generan solamente 100 tuplas nuevas.

*/

```
CREATE OR REPLACE PROCEDURE PR_RELLENAR (TUPLES IN NUMBER)
IS
    COD NUMBER;
    N    NUMBER;
    CAD VARCHAR2(15);
    DAY VARCHAR2(15);
    FEC DATE;
    N_LOOPS NUMBER := 100;
BEGIN
    IF TUPLES > 100 THEN
        N_LOOPS := TUPLES;
    END IF;
    FOR X IN 1..N_LOOPS LOOP
        COD := USUARIO2.SECUENCIA_DATOS.NEXTVAL;
        N    := NUM_ALEATORIO(2);
        CAD := CADENA_ALEATORIA(7);
        DAY := TO_CHAR(sysdate, 'DAY');
        FEC := CALCULAR_FECHA(SYSDATE, DAY, N);
        /*
        DBMS_OUTPUT.PUT_LINE('LOOP: ' || X);
        DBMS_OUTPUT.PUT_LINE('CODE: ' || COD);
        DBMS_OUTPUT.PUT_LINE('RAND_NUM: ' || N);
        DBMS_OUTPUT.PUT_LINE('RAND_STR: ' || CAD);
        DBMS_OUTPUT.PUT_LINE('DATE: ' || FEC);
        DBMS_OUTPUT.PUT_LINE(CHR(10));
        */
        INSERT INTO USUARIO2.DATOS VALUES(COD, CAD, FEC);
    END LOOP;
END RELLENAR;
```

```
GRANT EXECUTE ON USUARIO2.RELLENAR TO USUARIO2;
```

/*

```
SELECT * FROM USUARIO2.DATOS;
```

*/

/* 7

Cree una tabla llamada TB_OBJETOS con los siguientes atributos:

NOMBRE, CODIGO, FECHA_CREACION, FECHA_MODIFICACION, TIPO, ESQUEMA_ORIGINAL.

Recorra la vista ALL_OBJECTS y rellene esta tabla con los datos que se aportan en la vista.

Use un cursor y no un INSERT.

*/

```
CREATE TABLE USUARIO2.TB_OBJETOS (
```



```

NOMBRE VARCHAR2(30),
CODIGO NUMBER,
FECHA_CREACION DATE,
FECHA_MODIFICACION DATE,
TIPO VARCHAR2(30),
ESQUEMA_ORIGINAL NUMBER
);

CREATE OR REPLACE PROCEDURE PR_RELLENAR_OBJECTS
IS
    CURSOR C_OBJECTS IS SELECT OBJECT_NAME, OBJECT_ID, CREATED,
LAST_DDL_TIME, OBJECT_TYPE, NAMESPACE
FROM ALL_OBJECTS;
BEGIN
    FOR V_CURSOR IN C_OBJECTS LOOP
        INSERT INTO USUARIO2.TB_OBJETOS
            VALUES (V_CURSOR.OBJECT_NAME, V_CURSOR.OBJECT_ID,
V_CURSOR.CREATED, V_CURSOR.LAST_DDL_TIME, V_CURSOR.OBJECT_TYPE,
V_CURSOR.NAMESPACE);
    END LOOP;
END PR_RELLENAR_OBJECTS;

/* 8 */
/*
Cree una tabla TB_ESTILO con los siguientes atributos:
TIPO_OBJETO, PREFIJO.

En esta tabla se guardan unas normas de estilo de modo que a cada tipo de
objeto
le corresponde un prefijo en su identificador.

Así por ejemplo guardamos la tupla ('PROCEDURE','PR_') para indicar que un
nombre
correcto de procedimiento es PR_HOLA_MUNDO.
*/
CREATE TABLE USUARIO2.TB_ESTILO (
    TIPO_OBJETO VARCHAR2(30),
    PREFIJO VARCHAR2(20)
);

CREATE OR REPLACE PROCEDURE PR_RELLENAR_ESTILOS
IS
    CURSOR C_OBJECTS IS SELECT OBJECT_TYPE, CONCAT(SUBSTR(OBJECT_TYPE,1,2),
'_' ) AS PREFIX
FROM ALL_OBJECTS GROUP BY OBJECT_TYPE;
BEGIN
    FOR V_CURSOR IN C_OBJECTS LOOP
        INSERT INTO USUARIO2.TB_ESTILO VALUES (V_CURSOR.OBJECT_TYPE,
V_CURSOR.PREFIX);
    END LOOP;
END PR_RELLENAR_ESTILOS;

```



/* 9

Extienda el esquema de la tabla TB_OBJETOS en dos atributos:
ESTADO y NOMBRE_CORRECTO.

Cree un procedimiento llamado PR_COMPROBAR(ESQUEMA IN VARCHAR2) que recorre la
tabla TB_OBJETOS y comprueba si se cumplen las normas de estilo
según la tabla TB_ESTILO.

El parámetro que recibe es el identificador del esquema sobre
el que queremos comprobar las normas.
Si no se especifica, se comprueba en todos.

Actualice el atributo ESTADO de la tabla TB_OBJETOS con los valores
CORRECTO o INCORRECTO según las normas de estilo y el atributo
NOMBRE_CORRECTO con el
identificador con el prefijo adecuado.

El nuevo identificador se calcula anteponiendo el prefijo correcto al
identificador antiguo.
Si el identificador nuevo excede el tamaño del OBJECT_NAME de Oracle,
entonces pade el nuevo identificador por la derecha.

Use un cursor de actualización para realizar este procedimiento.

```
*/  
ALTER TABLE TB_OBJETOS ADD (  
    ESTADO VARCHAR2(30),  
    NOMBRE_CORRECTO VARCHAR2(30)  
);  
  
CREATE OR REPLACE PROCEDURE PR_COMPROBAR(ESQUEMA IN VARCHAR2)  
IS  
    CURSOR C_OBJECTS IS SELECT OBJECT_NAME, OBJECT_ID, CREATED,  
        LAST_DDL_TIME, OBJECT_TYPE, NAMESPACE, OWNER  
        FROM ALL_OBJECTS;  
  
    V_OBJECT_TYPE TB_ESTILO.TIPO_OBJETO%type;  
    ESTADO VARCHAR2(30);  
    NOMBRE_CORRECTO VARCHAR2(30);  
    PREFIJO VARCHAR2(30);  
BEGIN  
    FOR V_CURSOR IN C_OBJECTS LOOP  
        SELECT PREFIJO INTO V_OBJECT_TYPE FROM TB_ESTILO WHERE TIPO_OBJETO =  
V_CURSOR.OBJECT_TYPE;  
        PREFIJO := CONCAT(SUBSTR(V_CURSOR.OBJECT_NAME,1,2), '_');  
        NOMBRE_CORRECTO := RPAD(CONCAT(PREFIJO, V_CURSOR.OBJECT_NAME), 30);  
        IF ESQUEMA != '' THEN  
            IF V_CURSOR.OWNER = ESQUEMA THEN  
                IF V_OBJECT_TYPE = PREFIJO THEN  
                    ESTADO := 'CORRECTO';
```

```

        ELSE
            ESTADO := 'INCORRECTO';
        END IF;
        INSERT INTO USUARIO2.TB_OBJETOS
            VALUES(V_CURSOR.OBJECT_NAME, V_CURSOR.OBJECT_ID,
V_CURSOR.CREATED, V_CURSOR.LAST_DDL_TIME, V_CURSOR.OBJECT_TYPE,
V_CURSOR.NAMESPACE, ESTADO, NOMBRE_CORRECTO);
        END IF;
    ELSE
        IF V_OBJECT_TYPE = PREFIJO THEN
            ESTADO := 'CORRECTO';
        ELSE
            ESTADO := 'INCORRECTO';
        END IF;
        INSERT INTO USUARIO2.TB_OBJETOS
            VALUES(V_CURSOR.OBJECT_NAME, V_CURSOR.OBJECT_ID,
V_CURSOR.CREATED, V_CURSOR.LAST_DDL_TIME, V_CURSOR.OBJECT_TYPE,
V_CURSOR.NAMESPACE, ESTADO, NOMBRE_CORRECTO);
        END IF;

    END LOOP;
    COMMIT;
END PR_COMPROBAR;

```

/* 10

Cree una tabla llamada TB_ERRORES con los atributos FECHA, RUTINA, CODIGO y MENSAJE.

Cree varios procedimientos que producen errores de Oracle y guarda en la tabla TB_ERRORES un rastro de dichos errores.

Introduzca en estos procedimientos una sección de excepciones para introducir en TB_ERRORES

la aparición de este error:

ORA-01422: la recuperación exacta devuelve un número mayor de filas que el solicitado

*/

/* ERRORS https://www.techonthenet.com/oracle/exceptions/named_system.php */

```

CREATE TABLE TB_ERRORES (
    FECHA DATE,
    RUTINA VARCHAR2(30),
    CODIGO VARCHAR2(15),
    MENSAJE VARCHAR2(255)
);

```

```

CREATE PROCEDURE PR_SELECT_MAS_UNA_FILA
AS VAR_FILA ALL_TABLES%ROWTYPE;
BEGIN
    SELECT * INTO VAR_FILA FROM ALL_TABLES;

```

```

EXCEPTION
    WHEN TIMEOUT_ON_RESOURCE THEN
        INSERT INTO TB_ERRORES
            VALUES(SYSDATE, 'PR_SELECT_MAS_UNA_FILA', 'ORA-00051', 'You were
waiting for a resource and you timed out.');
```

waiting for a resource and you timed out.');

```

    WHEN TOO_MANY_ROWS THEN
        INSERT INTO TB_ERRORES
            VALUES(SYSDATE, 'PR_SELECT_MAS_UNA_FILA', 'ORA-01422', 'You tried
to execute a SELECT INTO statement and more than one row was returned.');
```

to execute a SELECT INTO statement and more than one row was returned.');

```

END PR_SELECT_MAS_UNA_FILA;

CREATE OR REPLACE PROCEDURE PR_RELLENAR2 (TUPLES IN NUMBER)
IS
    COD NUMBER;
    N    NUMBER;
    CAD VARCHAR2(15);
    DAY VARCHAR2(15);
    FEC DATE;
    N_LOOPS NUMBER := 100;
BEGIN
    IF TUPLES > 100 THEN
        N_LOOPS := TUPLES;
    END IF;
    FOR X IN 1..N_LOOPS LOOP
        COD := USUARIO2.SECUENCIA_DATOS.NEXTVAL;
        N    := NUM_ALEATORIO(2);
        CAD := CADENA_ALEATORIA(7);
        DAY := TO_CHAR(sysdate, 'DAY');
        FEC := CALCULAR_FECHA(SYSDATE, DAY, N);
        /*
        DBMS_OUTPUT.PUT_LINE('LOOP: ' || X);
        DBMS_OUTPUT.PUT_LINE('CODE: ' || COD);
        DBMS_OUTPUT.PUT_LINE('RAND_NUM: ' || N);
        DBMS_OUTPUT.PUT_LINE('RAND_STR: ' || CAD);
        DBMS_OUTPUT.PUT_LINE('DATE: ' || FEC);
        DBMS_OUTPUT.PUT_LINE(CHR(10));
        */
        INSERT INTO USUARIO2.DATOS VALUES(COD, CAD, FEC);
    END LOOP;
END RELLENAR;

```