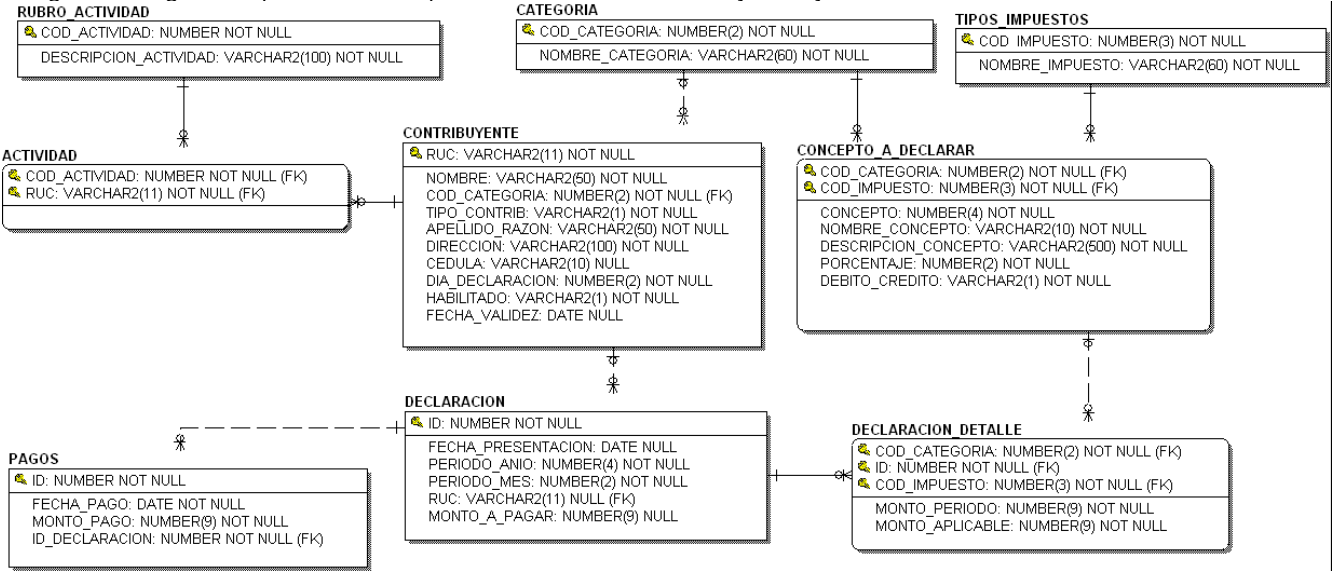


Carrera	CIENCIAS INFORMÁTICAS	Asignatura:	BASE DE DATOS 2	FECHA	PUNTAJE	Tema1	40		
Examen	SEGUNDO PARCIAL	Sección	SÁBADOS -J	12-11-2011		Tema2	20		
Profesor	Carmen N. Martínez W.	Nombre				Tema3	20		
Auxiliar:	Lilian Riveros	Alumno				TOTAL	80		

El siguiente diagrama representa un esquema del control de contribuyentes y declaraciones



### Tema 1: ( TOTAL: 40P)

#### a) Crear un tipo (35 P)

Cree el tipo T\_CONTRIBUYENTE como un objeto que posee los siguientes atributos:

- RUC
- NOMBRE
- APELLIDO\_RAZON
- TIPO\_CONTRIBUYENTE
- HABILITADO
- FECHA\_VALIDADEZ
- ACTIVIDADES

En donde ACTIVIDADES es del tipo TAB\_ACTIVIDAD. Y TAB\_ACTIVIDAD es una tabla con la lista de actividades o rubros del contribuyente (la descripción)

Tendrá también los siguientes métodos:

- El procedimiento P\_VERIFICAR\_HABILITACION que se aplicará sobre el objeto instanciado, realizando los siguientes pasos:
  - Verificar las declaraciones que pertenecen al RUC del objeto instanciado. La diferencia entre el mes y año de la última declaración **con pago realizado**, con respecto a la fecha del sistema no debe ser mayor a 1 mes (Ejemplo: Al mes de 11/2011, el **periodo de declaración** más reciente no puede ser anterior a 10/2011).
  - Si la última declaración fue hecha hace más de un mes, el atributo Habilitado quedará en ‘N’, de lo contrario quedará en ‘S’
- La función estática F\_OBTENER\_CONTRIBUYENTE, que reciba como parámetro el número de RUC y devuelva como parámetro un objeto del tipo T\_CONTRIBUYENTE con los datos asignados del contribuyente y su lista de actividades.

```
CREATE TYPE TAB_ACTIVIDAD IS TABLE OF VARCHAR2(100);
```

```
CREATE OR REPLACE TYPE T_CONTRIBUYENTE IS OBJECT
( RUC      VARCHAR2(11),
  NOMBRE   VARCHAR2(50),
  APELLIDO_RAZON VARCHAR2(50),
  TIPO_CONTRIBUYENTE VARCHAR2(1),
  HABILITADO VARCHAR2(1),
  FECHA_VALIDADEZ DATE,
  ACTIVIDADES TAB_ACTIVIDAD,
  MEMBER PROCEDURE P_VERIFICAR_HABILITACION,
  STATIC FUNCTION  F_OBTENER_CONTRIBUYENTE(PRUC VARCHAR2) RETURN T_CONTRIBUYENTE);
```

```
CREATE OR REPLACE TYPE BODY T_CONTRIBUYENTE IS
  MEMBER PROCEDURE P_VERIFICAR_HABILITACION IS
    V_FECHA DATE;
  BEGIN
  SELECT
```

Carrera	CIENCIAS INFORMÁTICAS	Asignatura:	BASE DE DATOS 2	FECHA	PUNTAJE	Tema1	40		
Examen	SEGUNDO PARCIAL	Sección	SÁBADOS -J	12-11-2011		Tema2	20		
Profesor	Carmen N. Martínez W.	Nombre				Tema3	20		
Auxiliar:	Lilian Riveros	Alumno				TOTAL	80		

```
MAX(TO_DATE(TO_CHAR(D.PERIODO_ANIO,'9999')|| TO_CHAR(D.PERIODO_MES,'09'),'YYYYMM')) INTO V_FECHA
FROM DECLARACION D, PAGOS P
WHERE P.ID = D.ID
AND D.RUC = SELF.RUC;
IF MONTHS_BETWEEN(SYSDATE, V_FECHA) >= 1 THEN
    SELF.HABILITADO := 'S';
ELSE
    SELF.HABILITADO := 'N';
END IF;
END;
STATIC FUNCTION F_OBTENER_CONTRIBUYENTE(PRUC VARCHAR2) RETURN T_CONTRIBUYENTE
IS
    CURSOR C_ACTIVIDAD IS
        SELECT RA.DESCRIPCION_ACTIVIDAD
        FROM ACTIVIDAD A, RUBRO_ACTIVIDAD RA
        WHERE RA.COD_ACTIVIDAD = A.COD_ACTIVIDAD
        AND A.RUC = PRUC;
    V_CONTRIBUYENTE CONTRIBUYENTE%ROWTYPE;
    V_CONTRIB T_CONTRIBUYENTE;
    V_ACTIVIDADES TAB_ACTIVIDAD := TAB_ACTIVIDAD();
    IND NUMBER(2) := 1;
BEGIN
    SELECT * INTO V_CONTRIBUYENTE
    FROM CONTRIBUYENTE
    WHERE RUC = PRUC;
    FOR REG IN C_ACTIVIDAD LOOP
        V_ACTIVIDADES.EXTEND;
        V_ACTIVIDADES(IND) := REG.DESCRIPCION_ACTIVIDAD;
        IND := IND + 1;
    END LOOP;
    V_CONTRIB := T_CONTRIBUYENTE(V_CONTRIBUYENTE.RUC, V_CONTRIBUYENTE.NOMBRE,
    V_CONTRIBUYENTE.APELLIDO_RAZON,V_CONTRIBUYENTE.TIPO_CONTRIB,
    V_CONTRIBUYENTE.HABILITADO, V_CONTRIBUYENTE.FECHA_VALIDEZ, V_ACTIVIDADES);
    RETURN V_CONTRIB;
END;
END;
```

**B) Crear tabla de BD (5P)**

Cree una tabla de **BD** denominada **NUEVO\_CONTRIBUYENTE** que tendrá una columna del tipo **T\_CONTRIBUYENTE**

```
CREATE TABLE NUEVO_CONTRIBUYENTE OF T_CONTRIBUYENTE
NESTED TABLE ACTIVIDADES STORE AS TABLA_ACTIVIDADES;
```

**Tema2 ( TOTAL 20 P):**

Escriba el o los triggers necesarios sobre la tabla **DECLARACIÓN\_DETALLE** para controlar lo siguiente:

- Al insertar un detalle de declaración:
  - Verificar que el código de categoría del concepto que se inserta, sea de la categoría a la que pertenece el contribuyente. En caso que así no sea, dará error.
  - Asignar el atributo aplicable que será igual al **MONTO\_PERIODO** por el porcentaje que le corresponde según el concepto
- No permita modificar ni borrar el registro

```
CREATE OR REPLACE TRIGGER T_BIUD_DECLA
BEFORE INSERT OR DELETE OR UPDATE ON DECLARACION_DETALLE
FOR EACH ROW
DECLARE
    V_COD CATEGORIA.COD_CATEGORIA%TYPE;
    V_PORC NUMBER(3);
BEGIN
    IF INSERTING THEN
        SELECT C.COD_CATEGORIA INTO V_COD
        FROM CONTRIBUYENTE C, DECLARACION D
        WHERE D.RUC = C.RUC
        AND D.ID = :NEW.ID;
        IF V_COD <> :NEW.COD_CATEGORIA THEN
            RAISE_APPLICATION_ERROR(-20000, 'LA CATEGORIA NO CORRESPONDE');
        END IF;
        SELECT PORCENTAJE INTO V_PORC
```

Carrera	CIENCIAS INFORMÁTICAS	Asignatura:	BASE DE DATOS 2	FECHA	PUNTAJE	Tema1	40		
Examen	SEGUNDO PARCIAL	Sección	SÁBADOS -J	12-11-2011		Tema2	20		
Profesor	Carmen N. Martínez W.	Nombre				Tema3	20		
Auxiliar:	Lilian Riveros	Alumno				TOTAL	80		

```
FROM CONCEPTO_A_DECLARAR
WHERE COD_IMPUESTO = :NEW.COD_IMPUESTO;
:NEW.MONTO_APLICABLE := ROUND (:NEW.MONTO_PERIODO * (V_PORC/100));
END IF;
IF DELETING OR UPDATING THEN
    RAISE_APPLICATION_ERROR(-20000, 'NO PUEDE BORRAR NI MODIFICAR' );
END IF;
END;
```

**Tema 3 (20 P):**

Por decisión de las autoridades, el atributo CEDULA y pasa a llamarse IUP(Identificador Único de Persona). Debe examinar las tablas de su esquema y verificar las que tienen columnas con el nombre CEDULA, y luego renombrar dicha columna a la denominación IUP. Desarrolle el procedimiento P\_RENOMBRAR que realice dicha tarea en todas las tablas QUE NO son propias del ejercicio del curso (debe excluir las tablas cuyo nombre empieza con ‘P\_’);

```
CREATE OR REPLACE PROCEDURE P_RENOMBRAR IS
    CURSOR C_CED IS
        SELECT TABLE_NAME FROM USER_TAB_COLUMNS
        WHERE COLUMN_NAME = 'CEDULA'
        AND TABLE_NAME NOT LIKE 'P\_%' ESCAPE '\';
    V_SENTENCIA VARCHAR2(500);
BEGIN
    FOR REG IN C_CED LOOP
        V_SENTENCIA := 'ALTER TABLE ' || REG.TABLE_NAME || ' RENAME COLUMN
CEDULA TO IUP';
        DBMS_OUTPUT.PUT_LINE(V_SENTENCIA);
        EXECUTE IMMEDIATE RTRIM(V_SENTENCIA);
    END LOOP;
END;
```