



UNIVERSIDADE
DE VIGO

ESCOLA SUPERIOR DE ENXEÑERÍA INFORMÁTICA

PRÁCTICA 8: TRIGGERS

GRUPO: BDII3_1

TEMÁTICA: HOTEL

DNI: 53821482X
DNI: 76735224W

NOMBRE: CERTAL PUGA, MANUEL
NOMBRE: GONZÁLEZ PEDROUZO,
ANTONIO ADRIÁN

DNI: 44657399R
DNI: 53817345J

NOMBRE: LÓPEZ MARCHANTE, NOA
NOMBRE: RAÑA GALLARDO, BREOGÁN

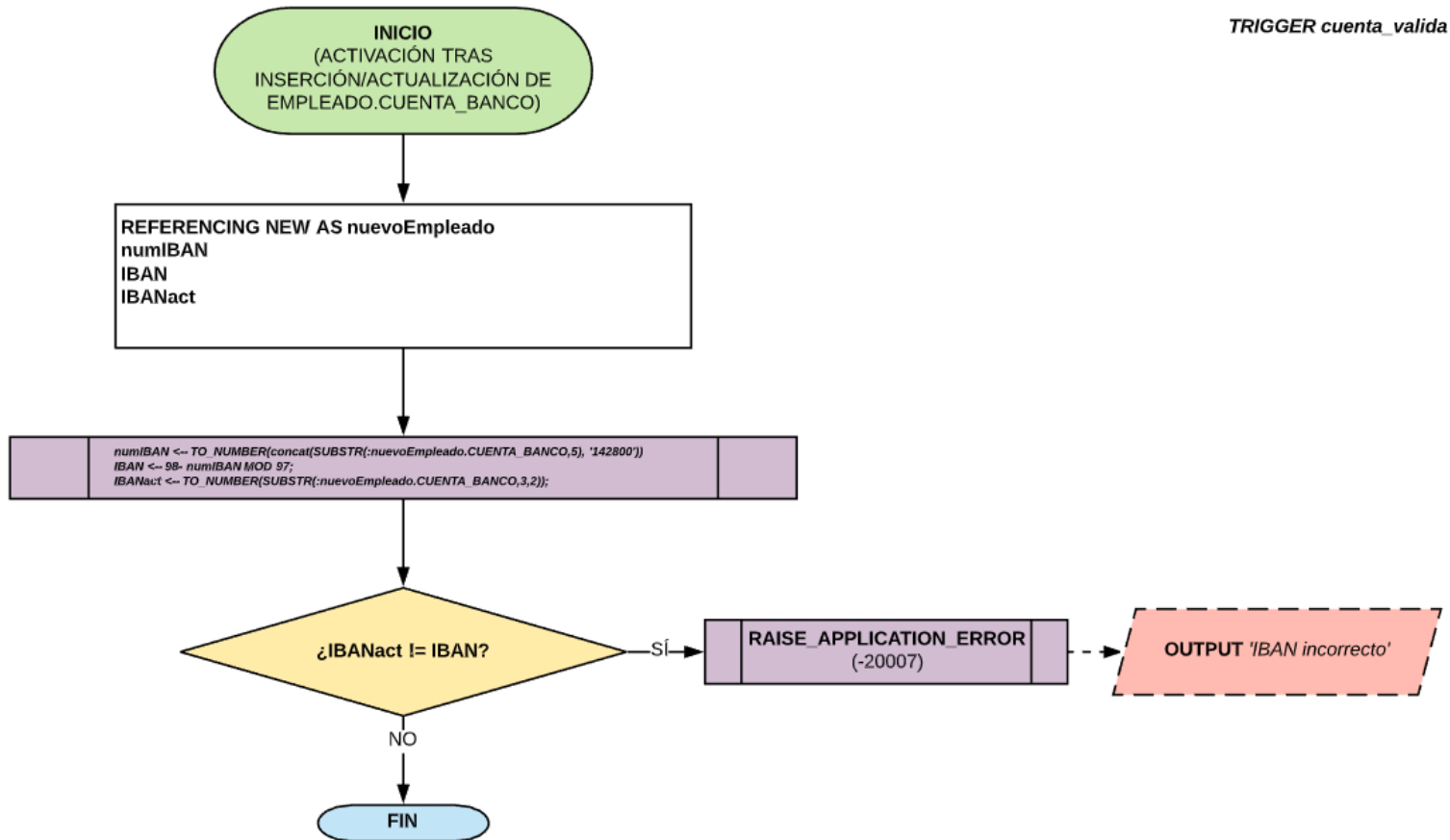


BASES DE DATOS II 2019-2020

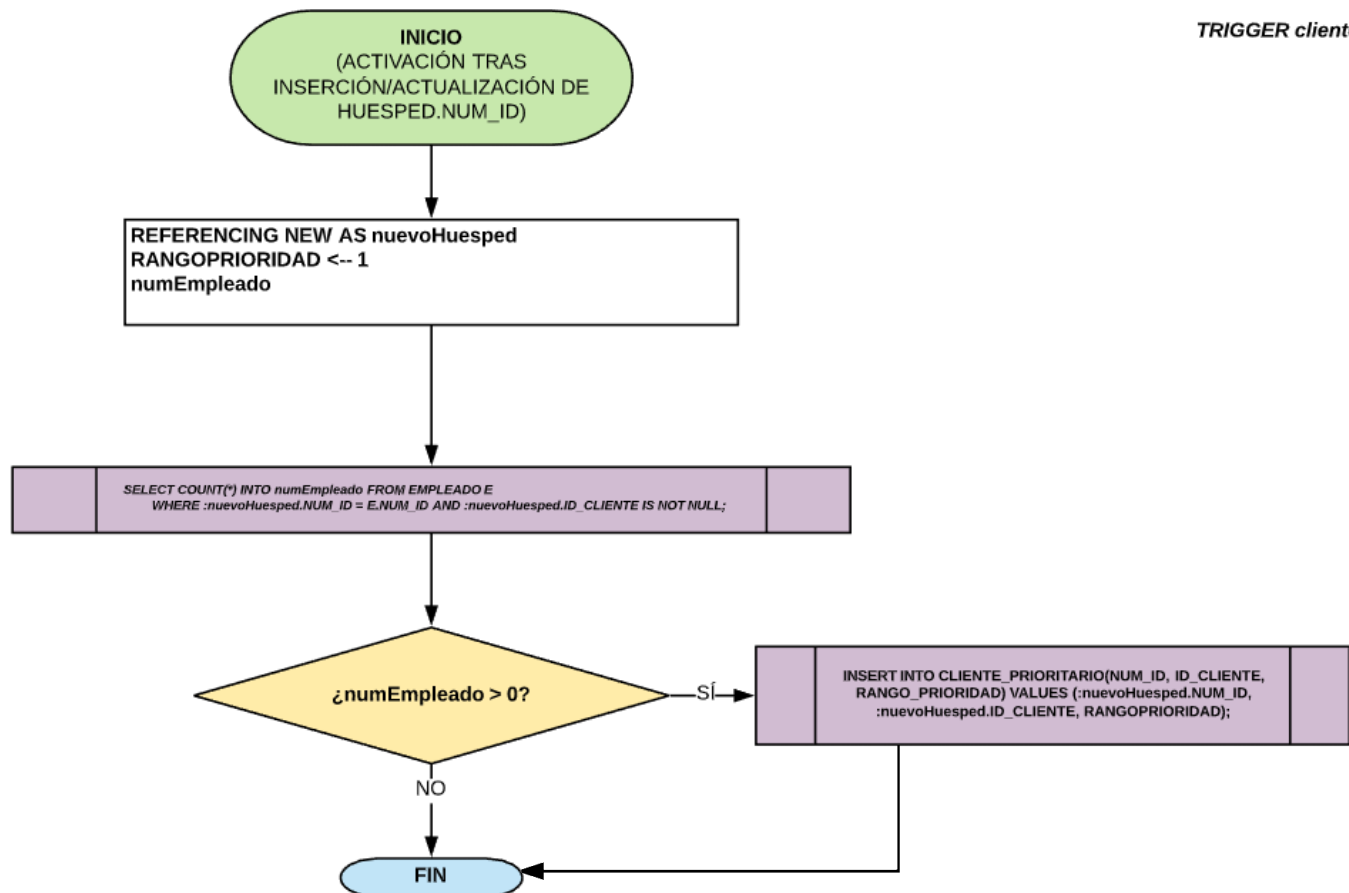
CALIFICACIÓN

DIAGRAMAS DE FLUJO DE TRIGGERS

TRIGGER cuenta_valida



TRIGGER cliente_prioritario_cliente



INICIO
(ACTIVACIÓN TRAS INSERCIÓN DE EMPLEADO)

REFERENCING NEW AS nuevoEmpleado
RANGOPRIORIDAD <-- 1
numHuesped
idCliente

SELECT COUNT(*) INTO numHuesped FROM HUESPED H
WHERE :nuevoEmpleado.NUM_ID = H.NUM_ID AND H.ID_CLIENTE IS NOT NULL;

¿numHuesped > 0?

Sí

NO

FIN

SELECT H.ID_CLIENTE INTO idCliente FROM HUESPED H
WHERE :nuevoEmpleado.NUM_ID = H.NUM_ID AND
H.ID_CLIENTE IS NOT NULL;

INSERT INTO CLIENTE_PRIORITARIO(NUM_ID,
ID_CLIENTE, RANGO_PRIORIDAD) VALUES
(:nuevoEmpleado.NUM_ID, idCliente, RANGOPRIORIDAD);

INICIO
(ACTIVACIÓN ANTES DE CADA
DECLARACIÓN DE BORRADO)

numHoteles

SELECT COUNT(*) INTO numHoteles FROM HOTEL;

INICIO II
(ACTIVACIÓN ANTES DEL BORRADO DE
CADA TUPLA AFECTADA)

¿numHoteles > 1?

Sí

NO

RAISE_APPLICATION_ERROR
(-20008)

OUTPUT 'La cadena hotelera
tiene que tener al menos un hotel.'

numHoteles <-- numHoteles - 1

FIN

SENTENCIAS DE DEFINICIÓN DE TRIGGERS

```

/*****
/* 8.- TRIGGERS PL/SQL */
*****/

--Validar Códigos Bancarios
CREATE OR REPLACE TRIGGER cuenta_valida
  AFTER INSERT OR UPDATE OF CUENTA_BANCO ON EMPLEADO
  REFERENCING NEW AS nuevoEmpleado
  FOR EACH ROW
DECLARE
  numIBAN      NUMBER;
  IBAN         NUMBER;
  IBANact      NUMBER;
BEGIN
  numIBAN := TO_NUMBER(concat(SUBSTR(:nuevoEmpleado.CUENTA_BANCO,5),
'142800'));
  IBAN := 98-MOD(numIBAN,97);
  IBANact := TO_NUMBER(SUBSTR(:nuevoEmpleado.CUENTA_BANCO,3,2));

  IF (IBANact != IBAN)
  THEN
    Raise_Application_Error(-20007, 'IBAN incorrecto');
  END IF;
END;
/
show errors

--Crear automáticamente cliente prioritario si se crea un huesped que
es cliente
-- y empleado
CREATE OR REPLACE TRIGGER cliente_prioritario_cliente
  AFTER INSERT OR UPDATE OF NUM_ID ON HUESPED
  REFERENCING NEW AS nuevoHuesped
  FOR EACH ROW
DECLARE
  RANGOPRIORIDAD      NUMBER DEFAULT 1;
  numEmpleado         NUMBER; --Si hay un empleado con el mismo ID y
es cliente, valdrá 1
BEGIN

  SELECT COUNT(*) INTO numEmpleado FROM EMPLEADO E
    WHERE :nuevoHuesped.NUM_ID = E.NUM_ID AND
:nuevoHuesped.ID_CLIENTE IS NOT NULL;
```

```

        IF (numEmpleado > 0)
        THEN
            INSERT INTO CLIENTE_PRIORITARIO(NUM_ID, ID_CLIENTE,
RANGO_PRIORIDAD) VALUES (:nuevoHuesped.NUM_ID,
:nuevoHuesped.ID_CLIENTE, RANGOPRIORIDAD);
        END IF;
    END;
/
show errors

--Crear automáticamente cliente prioritario si se crea un empleado que
es huesped
-- y cliente
CREATE OR REPLACE TRIGGER cliente_prioritario_empleado
    AFTER INSERT ON EMPLEADO
    REFERENCING NEW AS nuevoEmpleado
    FOR EACH ROW
DECLARE
    RANGOPRIORIDAD        NUMBER DEFAULT 1;
    numHuesped             NUMBER; --Si hay un huésped con el mismo ID y
es cliente, valdrá 1
    idCliente             HUESPED.NUM_ID%TYPE;
BEGIN

    SELECT COUNT(*) INTO numHuesped FROM HUESPED H
        WHERE :nuevoEmpleado.NUM_ID = H.NUM_ID AND H.ID_CLIENTE IS NOT
NULL;

    IF (numHuesped > 0)
    THEN
        SELECT H.ID_CLIENTE INTO idCliente FROM HUESPED H
            WHERE :nuevoEmpleado.NUM_ID = H.NUM_ID AND H.ID_CLIENTE IS
NOT NULL;

        INSERT INTO CLIENTE_PRIORITARIO(NUM_ID, ID_CLIENTE,
RANGO_PRIORIDAD) VALUES (:nuevoEmpleado.NUM_ID, idCliente,
RANGOPRIORIDAD);
    END IF;
END;
/
show errors

-- Impedir que la cadena hotelera se quede sin hoteles.
CREATE OR REPLACE TRIGGER min_hotel_cadenaHotelera

```

```

FOR DELETE ON HOTEL
COMPOUND TRIGGER
    numHoteles    NUMBER;
BEFORE STATEMENT IS
BEGIN
    SELECT COUNT(*) INTO numHoteles FROM HOTEL;
END BEFORE STATEMENT;

BEFORE EACH ROW IS
BEGIN
    IF numHoteles > 1 THEN
        numHoteles := NumHoteles - 1;
    ELSE
        Raise_Application_Error(-20008, 'La cadena hotelera tiene que
tener al menos un hotel');
    END IF;
END BEFORE EACH ROW;
END;
/
show errors

```

SENTENCIAS DE PRUEBA DE TRIGGERS

```
/* 10.- Bloque para prueba de Triggers */
```

```
SELECT * FROM CLIENTE_PRIORITARIO; --tabla clientes prioritarios antes de pruebas de Triggers
```

```
SELECT * FROM HOTEL; --tabla hotel antes de pruebas de Triggers
```

```
SET SERVEROUTPUT ON
```

```
BEGIN
```

```
    DBMS_OUTPUT.NEW_LINE;
```

```
    -- PRUEBA Trigger cuenta_valida CASO EN EL QUE GENERA ERROR
```

```
    BEGIN
```

```
        DBMS_OUTPUT.NEW_LINE;
```

```
        DBMS_OUTPUT.PUT_LINE('=====>PRUEBA DEL TRIGGER cuenta_valida [FALLARÁ] (ANTES)');
```

```
        DBMS_OUTPUT.PUT_LINE(' Intentamos introducir una persona...');
```

```
        INSERT INTO PERSONA(NUM_ID, NOMBRE, TELEFONO, FECHA_NACIMIENTO) VALUES ('71524488A', 'Marcos Rodríguez López', '654221356', '20/02/1992');
```

```
        DBMS_OUTPUT.PUT_LINE(' Conseguimos introducir la persona');
```

```
        DBMS_OUTPUT.PUT_LINE(' Intentamos introducir un empleado con un IBAN incorrecto...');
```

```
        INSERT INTO EMPLEADO(NUM_ID, ESPECIALIDAD, CUENTA_BANCO, ANHOS_EXPERIENCIA, ID_HOTEL, ID_PRACTICA_SUPERVISADA) VALUES ('71524488A', 'ATENCIÓN AL PÚBLICO', 'ES3100315882373748044941', 4, 'HOU220', NULL);
```

```
        DBMS_OUTPUT.PUT_LINE(' Conseguimos introducir el empleado (Fallo de trigger!); -- esta linea no se muestra si el trigger funciona adecuadamente
```

```
        DBMS_OUTPUT.PUT_LINE('=====>PRUEBA DEL TRIGGER cuenta_valida (DESPUÉS)'); -- este texto no se visualizará porque se produce una excepción antes
```

```
    EXCEPTION
```

```
        WHEN OTHERS THEN
```

```
            DBMS_OUTPUT.PUT_LINE('[EXCEPCIÓN]');
```

```
            DBMS_OUTPUT.PUT_LINE('[Código]: ' || SQLCODE);
```

```
            DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SUBSTR(SQLERRM, 11, 100));
```

```
    END;
```

```

-- PRUEBA Trigger cuenta_valida CASO EN EL QUE NO GENERA ERROR
BEGIN
    DBMS_OUTPUT.NEW_LINE;

    DBMS_OUTPUT.PUT_LINE('=====>PRUEBA DEL TRIGGER [FUNCIONARÁ]
cuenta_valida (ANTES)');
    DBMS_OUTPUT.PUT_LINE(' Intentamos introducir un empleado con
un IBAN correcto...');
    INSERT INTO EMPLEADO(NUM_ID, ESPECIALIDAD, CUENTA_BANCO,
ANHOS_EXPERIENCIA, ID_HOTEL, ID_PRACTICA_SUPERVISADA) VALUES
('71524488A', 'ATENCIÓN AL PÚBLICO', 'ES3100315882373728044941', 4,
'HOU220', NULL);
    DBMS_OUTPUT.PUT_LINE(' Conseguimos introducir el empleado');
    DBMS_OUTPUT.PUT_LINE('=====>PRUEBA DEL TRIGGER cuenta_valida
(DESPUÉS)');
    EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('[EXCEPCIÓN]');
        DBMS_OUTPUT.PUT_LINE('[Código]: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SUBSTR(SQLERRM, 11,
100));
    END;

-- PRUEBA Trigger cliente_prioritario_empleado
BEGIN
    DBMS_OUTPUT.NEW_LINE;

    DBMS_OUTPUT.PUT_LINE('=====>PRUEBA DEL TRIGGER
cliente_prioritario_empleado (ANTES)');
    DBMS_OUTPUT.PUT_LINE(' Intentamos introducir un empleado
que ya es huésped y cliente...');
    INSERT INTO EMPLEADO(NUM_ID, ESPECIALIDAD, CUENTA_BANCO,
ANHOS_EXPERIENCIA, ID_HOTEL, ID_PRACTICA_SUPERVISADA) VALUES
('23465138Z', 'ATENCIÓN AL PÚBLICO', 'ES1701288369169937338266', 2,
'HOU220', NULL);
    DBMS_OUTPUT.PUT_LINE(' Conseguimos introducir el empleado
(Visualizar tabla clientes prioritarios al final de la ejecución.)');
    DBMS_OUTPUT.PUT_LINE('=====>PRUEBA DEL TRIGGER
cliente_prioritario_empleado (DESPUÉS)');
    EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('[EXCEPCIÓN]');
        DBMS_OUTPUT.PUT_LINE('[Código]: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SUBSTR(SQLERRM, 11,
100));
    END;

```



```

-- PRUEBA Trigger cliente_prioritario_cliente
BEGIN
    DBMS_OUTPUT.NEW_LINE;

    DBMS_OUTPUT.PUT_LINE('=====>PRUEBA DEL TRIGGER
cliente_prioritario_cliente (ANTES)');
    DBMS_OUTPUT.PUT_LINE(' Intentamos introducir un cliente y
huesped que ya es empleado...');
    INSERT INTO CLIENTE(ID_CLIENTE, FECHA_PRIMER_CONTRATO) VALUES
('CLI01234', '08/10/2015');
    INSERT INTO HUESPED(NUM_ID, ID_CLIENTE, NACIONALIDAD,
NUM_TARJETA) VALUES
('65489481B', 'CLI01234', 'España', '557789332312894567');
    DBMS_OUTPUT.PUT_LINE(' Conseguimos introducirlo (Visualizar
tabla clientes prioritarios al final de la ejecución.));
    DBMS_OUTPUT.PUT_LINE('=====>PRUEBA DEL TRIGGER
cliente_prioritario_cliente (DESPUÉS)');
    EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('[EXCEPCIÓN]');
        DBMS_OUTPUT.PUT_LINE('[Código]: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SUBSTR(SQLERRM, 11,
100));
    END;

-- PRUEBA Trigger min_hotel_cadenaHotelera
BEGIN
    DBMS_OUTPUT.NEW_LINE;

    DBMS_OUTPUT.PUT_LINE('=====>PRUEBA DEL TRIGGER
min_hotel_cadenaHotelera (ANTES)');
    DBMS_OUTPUT.PUT_LINE(' Intentamos borrar todos los hoteles de
la cadena....');
    DELETE FROM HOTEL;
    DBMS_OUTPUT.PUT_LINE(' Lo conseguimos (fallo en el
trigger!)); -- A partir de esta línea, no debería ejecutarse si el
trigger funciona.
    DBMS_OUTPUT.PUT_LINE('=====>PRUEBA DEL TRIGGER
min_hotel_cadenaHotelera (DESPUÉS)');
    EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('[EXCEPCIÓN]');
        DBMS_OUTPUT.PUT_LINE('[Código]: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SUBSTR(SQLERRM, 11,
100));
    END;

```

END;

/

SELECT * FROM CLIENTE_PRIORITARIO; --tabla clientes prioritarios tras pruebas de Triggers

SELECT * FROM HOTEL; --tabla hotel tras pruebas de Triggers

HOJA DE FIRMAS

DNI: 53821482X

NOMBRE: CERTAL PUGA, MANUEL

DNI: 76735224W

**NOMBRE: GONZÁLEZ PEDROUZO,
ANTONIO ADRIÁN**

DNI: 44657399R

NOMBRE: LÓPEZ MARCHANTE, NOA

DNI: 53817345J

NOMBRE: RAÑA GALLARDO, BREOGÁN