

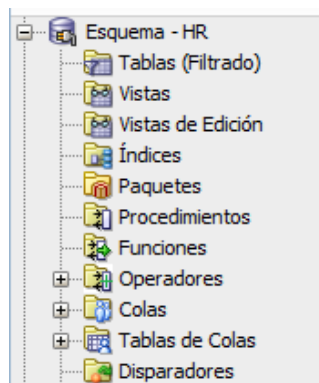
Ciclo Formativo GRADO SUPERIOR:	DESARROLLO DE APLICACIONES MULTIPLATAFORMA	IFCS02
Módulo Profesional Clave: 01	BASES DE DATOS	
Nº de Expediente:	Nif: 52018081Q	Fecha: 27/5/2022
Nombre y Apellidos: Daniel Izquierdo Bonilla		

- Este examen se entrega en único fichero de texto (word/odt/pdf) en el aula virtual **a partir de esta plantilla**. Durante este examen no está permitido el uso de internet en el PC (excepto para la descarga de los scripts y la entrega del doc) ni el uso de móviles (deben estar bien guardados) u otro tipo de dispositivos.
- Se usará el PC solo para ejecutar SQL-DEVELOPER. Se permite también el uso de un prontuario en papel (1 folio).
- Se debe guardar absoluto silencio y no está permitida ninguna interacción con el resto de alumnos. Si tienes alguna duda o necesidad, levanta la mano y el profesor te atenderá en cuanto pueda.
- Lee con detenimiento y atención este documento y las preguntas, si hace falta varias veces. Tómate tu tiempo (hay suficiente y de sobra para completarlo) y completa las preguntas en esta misma plantilla en formato electrónico.
- Pon el nombre en la cabecera del documento y nombra el documento final como APELLIDO_nombre.doc

PREPARACIÓN DEL ENTORNO:

Trabajaremos con el esquema HR, Se encuentran en el aula virtual los ficheros de script **HR_0.dropFP.sql**, **HR_1.drop.sql**, **HR_2.cre.sql**, **HR_3.popul.sql** y **HR_4.idx.sql** de borrado, creación de tablas, población de datos e índices sobre ese esquema HR.

IMPORTANTE: tras ejecutar las sentencias de DROP de los dos primeros ficheros **se debe comprobar que se han borrado cualquier otra tabla, vista, función, procedimiento o trigger previamente existente** de tal manera que no quede ninguno previo a la creación de tablas y población: (Pueden borrarse con el SQLDEVELOPER usando el botón derecho), debiendo quedar así el entorno;



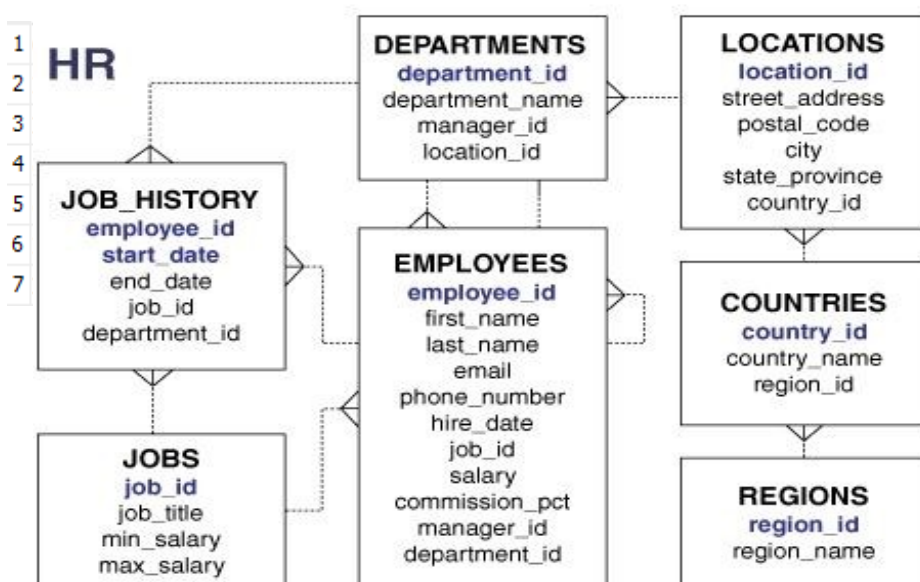
El profesor comprobará que todo queda borrado y podrá proseguirse.

Una vez borrados ejecutaremos los scripts **HR_2.cre.sql** y **HR_3.popul.sql** de creación de tablas población respectivamente y finalmente **HR_4.idx.sql** para los índices. Para comprobar que la carga ha sido correcta podemos ejecutar la siguiente sentencia que nos dará el número de filas de cada tabla:

```
SELECT TABLE_NAME, num_rows from user_tables order by table_name;
```

Puede comprobarse si coincide con la lista de cuenta de registros.

Como ayuda, se muestra el modelo de HR:



TAREAS: (en cada parte se anexará texto con el código y captura con resultados de ejemplo). Es importante incluir en el texto incluir un comentario con el nombre del autor. También en la captura que se vea claramente que el programa ha compilado correctamente.

1. **BLOQUE ANÓNIMO SENCILLO** (1 punto)

Hacer un bloque anónimo que imprima el nombre del país (country_name) con más departamentos

```
-- Ej1
-- Daniel Izquierdo Bonilla
set serveroutput on
DECLARE
    cuenta INTEGER;
    pais    COUNTRIES.COUNTRY_NAME%type;
BEGIN
    SELECT COUNTRIES.COUNTRY_NAME, COUNT(DEPARTMENT_ID) AS NUM_DEPARTAMENTOS
    INTO pais,cuenta
    FROM COUNTRIES
        JOIN LOCATIONS on COUNTRIES.COUNTRY_ID = LOCATIONS.COUNTRY_ID
        JOIN DEPARTMENTS on LOCATIONS.LOCATION_ID = DEPARTMENTS.LOCATION_ID
    group by COUNTRIES.COUNTRY_NAME
    order by NUM_DEPARTAMENTOS desc fetch first 1 rows only;

    DBMS_OUTPUT.PUT_LINE('Pais: ' || pais || ' | Numero de Departamentos: ' || cuenta);
end;
```

```
SELECT COUNTRIES.COUNTRY_NAME, COUNT(DEPARTMENT_ID) AS NUM_DEPARTAMENTOS
INTO pais,cuenta
FROM COUNTRIES
    JOIN LOCATIONS on COUNTRIES.COUNTRY_ID = LOCATIONS.COUNTRY_ID
    JOIN DEPARTMENTS on LOCATIONS.LOCATION_ID = DEPARTMENTS.LOCATION_ID
group by COUNTRIES.COUNTRY_NAME
order by NUM_DEPARTAMENTOS desc fetch first 1 rows only;

DBMS_OUTPUT.PUT_LINE('Pais: ' || pais || ' | Numero de Departamentos: ' || cuenta);
end;
[2022-05-27 13:15:22] completed in 10 ms
Pais: United States of America | Numero de Departamentos: 23
```

2. **TRATAMIENTO DE EXCEPCIONES** (1.5 puntos)

Hacer un bloque anónimo que pida un id de empleado y devuelva su nombre, apellidos, salario e id del departamento. Debe gestionar las siguientes excepciones:

- Que no exista el empleado con ese id (interna)
- Que no tenga departamento (definida) → debe dar aviso pero mostrar datos
- Que no tenga jefe (definida) → debe dar aviso pero NO mostrar datos
- Otras (OTHERS)

```
-- Ej2
-- Daniel Izquierdo Bonilla
set serveroutput on
DECLARE
    idEmp    EMPLOYEES.EMPLOYEE_ID%type;
    nom      EMPLOYEES.FIRST_NAME%type;
```

```
ape      EMPLOYEES.LAST_NAME%type;
salario  EMPLOYEES.SALARY%type;
idDep    EMPLOYEES.DEPARTMENT_ID%type;
cuenta   INTEGER;

no_deparment EXCEPTION;
no_boss   EXCEPTION;
BEGIN
    idEmp := &dime_id;

    SELECT COUNT(DEPARTMENT_ID) AS cue INTO cuenta FROM EMPLOYEES WHERE EMPLOYEE_ID =
idEmp;

    if cuenta = 0 then
        raise no_deparment;
    end if;

    SELECT COUNT(MANAGER_ID) AS cue INTO cuenta FROM EMPLOYEES WHERE EMPLOYEE_ID =
idEmp;

    if cuenta = 0 then
        raise no_boss;
    end if;

    SELECT FIRST_NAME, LAST_NAME, SALARY, DEPARTMENT_ID
    INTO nom, ape, salario, idDep
    FROM EMPLOYEES
    WHERE EMPLOYEE_ID = idEmp;

    DBMS_OUTPUT.PUT_LINE('Nombre: ' || nom || ' | Apellido: ' || ape || ' | Salario: '
|| salario ||
                        ' | ID del Departamento: ' || idDep);

EXCEPTION
    when NO_DATA_FOUND then DBMS_OUTPUT.PUT_LINE('Empleado no econtrado');
    when no_deparment then DBMS_OUTPUT.PUT_LINE('El empleado con la ID introducida no
tiene departamento');
    when no_boss then dbms_output.put_line('El empleado con la ID introducida no tiene
manager');
    when others then dbms_output.put_line('Error inesperado');
end;
```

```
DBMS_OUTPUT.PUT_LINE('Nombre: ' || nom || ' | Apellido: ' || ape || ' | Salario: ' || salario ||
                        ' | ID del Departamento: ' || idDep);

EXCEPTION
    when NO_DATA_FOUND then DBMS_OUTPUT.PUT_LINE('Empleado no econtrado');
    when no_deparment then DBMS_OUTPUT.PUT_LINE('El empleado con la ID introducida no tiene departamento');
    when no_boss then dbms_output.put_line('El empleado con la ID introducida no tiene manager');
    when others then dbms_output.put_line('Error inesperado');
end;
[2022-05-27 13:16:31] completed in 12 ms
Nombre: John | Apellido: Chen | Salario: 8200 | ID del Departamento: 100
```

3. CURSORES (1.5 puntos)

Hacer un bloque anónimo que muestre el nombre, ciudad y apellido del jefe de los departamentos cuya ciudad se pide por teclado. El cursor debe declararse, abrirse, usar FETCH y cerrarse (es decir, no se permite FOR en este caso)

```
-- Ej3
-- Daniel Izquierdo Bonilla
set serveroutput on
declare
    ciudadUser LOCATIONS.CITY%type;
    ciudad     LOCATIONS.CITY%type;
    nom        EMPLOYEES.FIRST_NAME%type;
    ape        EMPLOYEES.LAST_NAME%type;
    dep        DEPARTMENTS.DEPARTMENT_NAME%type;
    cursor c_ej is select EMPLOYEES.FIRST_NAME, LOCATIONS.CITY, EMPLOYEES.LAST_NAME,
DEPARTMENT_NAME
                        from EMPLOYEES
                        join DEPARTMENTS on EMPLOYEES.DEPARTMENT_ID =
DEPARTMENTS.DEPARTMENT_ID
                        join LOCATIONS on DEPARTMENTS.LOCATION_ID =
LOCATIONS.LOCATION_ID
                        where CITY = ciudadUser and EMPLOYEE_ID = DEPARTMENTS.MANAGER_ID;
begin
    ciudadUser := '&dime_ciudad';

    open c_ej;
    fetch c_ej into nom, ciudad, ape, dep;
    while c_ej%found
        loop
            dbms_output.put_line('Nombre: ' || nom || ' | Ciudad: ' || ciudad || ' |
Apellido: ' ||
                                ape || ' | Departamento: ' || dep);
            fetch c_ej into nom, ciudad, ape, dep;
        end loop;
    close c_ej;
end;
```

```
    open c_ej;
    fetch c_ej into nom, ciudad, ape, dep;
    while c_ej%found
        loop
            dbms_output.put_line('Nombre: ' || nom || ' | Ciudad: ' || ciudad || ' | Apellido: ' ||
                                ape || ' | Departamento: ' || dep);
            fetch c_ej into nom, ciudad, ape, dep;
        end loop;
    close c_ej;
end;
[2022-05-27 13:18:00] completed in 18 ms
Nombre: Susan | Ciudad: London | Apellido: Mavris | Departamento: Human Resources
```

4. FUNCIONES (1 punto)

(no usa la BD HR)

Hacer una función llamada INVIERTE(texto) que invierta una cadena. P.ej. INVIERTE(ejemplo) = olpmeje (No pueden usarse funciones pre-hechas, aparte de SUBSTR)

```
-- Ej4
-- Daniel Izquierdo Bonilla
create or replace function INVIERTE(texto VARCHAR2) return varchar2
is
  cuenta INTEGER;
  aux    VARCHAR2(100);
begin
  aux := '';
  cuenta := length(texto);

  loop
    aux := aux || substr(texto, cuenta, 1);
    cuenta := cuenta - 1;
    if cuenta = 0 then
      return aux;
    end if;
  end loop;
end;
```

```
    aux := '';
    cuenta := length(texto);

    loop
      aux := aux || substr(texto, cuenta, 1);
      cuenta := cuenta - 1;
      if cuenta = 0 then
        return aux;
      end if;
    end loop;

  end;
[2022-05-27 13:18:38] completed in 4 ms
```

```
--LISTA_SEDES('United States of America');
--CADENA_JEFES(206);

end;
[2022-05-27 13:19:25] completed in 5 ms
leinad
HR> begin
  DBMS_OUTPUT.PUT_LINE(INVIERTE('daniel'));
  --LISTA_SEDES('United States of America');
  --CADENA_JEFES(206);

end;
[2022-05-27 13:19:25] completed in 5 ms
leinad
```

5. **PROCEDIMIENTOS** (1.75 puntos)

Hacer un procedimiento LISTA_SEDES (pais) que liste la dirección (street_adress) y el nº de departamentos de cada una de las sedes (incluyendo sedes vacías) de un determinado país (id), que será su parámetro. (Puede emplearse FOR) (Se recomienda probar la SELECT previamente)

```
-- Ej5
-- Daniel Izquierdo Bonilla
CREATE OR REPLACE PROCEDURE LISTA_SEDES(pais COUNTRIES.COUNTRY_NAME%TYPE) IS
BEGIN
    FOR registro IN (SELECT LOCATIONS.STREET_ADDRESS as direc, COUNT(DEPARTMENT_ID) AS
dep
                        FROM DEPARTMENTS
                        Right Join LOCATIONS ON DEPARTMENTS.LOCATION_ID =
LOCATIONS.LOCATION_ID
                        JOIN COUNTRIES on COUNTRIES.COUNTRY_ID =
LOCATIONS.COUNTRY_ID
                        WHERE COUNTRIES.COUNTRY_NAME = pais
                        GROUP BY LOCATIONS.STREET_ADDRESS)
    LOOP
        DBMS_OUTPUT.PUT_LINE('Direccion: ' || registro.direc || ' | Departamentos: '
|| registro.dep);
    END LOOP;
END LISTA_SEDES;
```

```
BEGIN
    FOR registro IN (SELECT LOCATIONS.STREET_ADDRESS as direc, COUNT(DEPARTMENT_ID) AS dep
                        FROM DEPARTMENTS
                        Right Join LOCATIONS ON DEPARTMENTS.LOCATION_ID = LOCATIONS.LOCATION_ID
                        JOIN COUNTRIES on COUNTRIES.COUNTRY_ID = LOCATIONS.COUNTRY_ID
                        WHERE COUNTRIES.COUNTRY_NAME = pais
                        GROUP BY LOCATIONS.STREET_ADDRESS)
    LOOP
        DBMS_OUTPUT.PUT_LINE('Direccion: ' || registro.direc || ' | Departamentos: ' || registro.dep);
    END LOOP;
END LISTA_SEDES;
[2022-05-27 13:20:12] completed in 7 ms
```

```
[2022-05-27 13:19:25] completed in 5 ms
leinad
HR> begin
    --DBMS_OUTPUT.PUT_LINE(INVIERTE('daniel'));
    LISTA_SEDES('United States of America');
    --CADENA_JEFES(206);
end;
[2022-05-27 13:20:49] completed in 6 ms
Direccion: 2007 Zagora St | Departamentos: 0
Direccion: 2014 Jabberwocky Rd | Departamentos: 1
Direccion: 2011 Interiors Blvd | Departamentos: 1
Direccion: 2004 Charade Rd | Departamentos: 21
```

6. **TRIGGERS** (1.75 puntos)

Antes, crear (DDL) una tabla LOGGING con los campos: NOMBRE, APELLIDO, TIPO_CAMBIO, ANTIGUO, NUEVO (Todos VARCHAR). Hacer un trigger que ante cualquier cambio de email de un empleado inserte un registro en la tabla LOGGING con los valores correspondientes. ADEMÁS debe también insertar un registro en el caso de que se aumente el salario en más de un 20%. TIPO_CAMBIO será “Aumento excesivo” o “Cambio email” en cada caso. ANTIGUO y NUEVO serán los valores antiguos y nuevos de los cambios

(En este ejercicio debe anexarse además del código del trigger, la captura del select * de la tabla LOGGING)

```
-- Ej6
-- Daniel Izquierdo Bonilla
CREATE TABLE LOGGING
(
    NOMBRE          VARCHAR2(100),
    APELLIDO        VARCHAR2(100),
    TIPO_CAMBIO     VARCHAR2(100),
    ANTIGUO         VARCHAR2(100),
    NUEVO           VARCHAR2(100)
);

CREATE OR REPLACE TRIGGER CONTROL_CAMBIOS
BEFORE UPDATE
ON EMPLOYEES
FOR EACH ROW
BEGIN
    IF UPDATING ('SALARY') THEN
        IF :new.SALARY > :old.SALARY * 1.2 THEN
            INSERT INTO LOGGING VALUES (:new.FIRST_NAME, :new.LAST_NAME, 'Aumento
excesivo salario', :old.SALARY, :new.SALARY);
        end if;
    end if;

    IF UPDATING ('EMAIL') THEN
        INSERT INTO LOGGING
VALUES (:new.FIRST_NAME, :new.LAST_NAME, 'Cambio email', :old.EMAIL,
:new.EMAIL);
    END IF;
END;
```

```
IF UPDATING ('SALARY') THEN
    IF :new.SALARY > :old.SALARY * 1.2 THEN
        INSERT INTO LOGGING VALUES (:new.FIRST_NAME, :new.LAST_NAME, 'Aumento excesivo salario', :old.SALARY, :new.SALARY);
    end if;
end if;

IF UPDATING ('EMAIL') THEN
    INSERT INTO LOGGING
VALUES (:new.FIRST_NAME, :new.LAST_NAME, 'Cambio email', :old.EMAIL, :new.EMAIL);
END IF;
END;
[2022-05-27 12:45:39] completed in 26 ms
```

	NOMBRE	APELLIDO	TIPO_CAMBIO	ANTIGUO	NUEVO
1	Trenna	Rajs	Aumento excesivo salario	3500	5500
2	Trenna	Rajs	Cambio email	TRAJS	a@a

7. COMBINADO (1.5 puntos)

Hacer un procedimiento CADENA_JEFES(id) que dado el id de un empleado, devuelva un listado de la cadena de mando de ese empleado hasta el CEO (Steven King). Cada línea debe imprimir el nombre y apellido de cada jefe. Debe de tratar la excepción de que ese id no exista. HINTS: puede hacerse de manera recursiva. El caso del CEO puede hacerse como excepción. Se adjunta captura de ejemplo para el empleado 206.

NOTA1: Un código con errores de compilación no puntúa.
NOTA2: No se permite mirar el trabajo de los compañeros.

William Gietz tiene como jefe a Shelley Higgins Shelley Higgins tiene como jefe a Neena Kochhar Neena Kochhar tiene como jefe a Steven King Steven King es el CEO
--

```
-- Ej7
-- Daniel Izquierdo Bonilla
CREATE OR REPLACE PROCEDURE CADENA_JEFES(idEmp EMPLOYEES.EMPLOYEE_ID%TYPE) IS
    nom    EMPLOYEES.FIRST_NAME%type;
    ape    EMPLOYEES.LAST_NAME%type;
    man    EMPLOYEES.MANAGER_ID%type;
    auxNom EMPLOYEES.FIRST_NAME%type;
    auxApe EMPLOYEES.LAST_NAME%type;
BEGIN
    SELECT FIRST_NAME, LAST_NAME, MANAGER_ID
    INTO nom,ape,man
    FROM EMPLOYEES
    WHERE EMPLOYEE_ID = idEmp;

    if man is null then
        DBMS_OUTPUT.PUT_LINE(nom || ' ' || ape || ' es el CEO');
    elsif man is not null then
        SELECT FIRST_NAME, LAST_NAME
        INTO auxNom,auxApe
        FROM EMPLOYEES
        WHERE EMPLOYEE_ID = man;
        DBMS_OUTPUT.PUT_LINE(nom || ' ' || ape || ' tiene como jefe a ' || auxNom || ' '
|| auxApe);
        CADENA_JEFES(man);
    end if;
END CADENA_JEFES;

DBMS_OUTPUT.PUT_LINE(nom || ' ' || ape || ' es el CEO');
elsif man is not null then
    SELECT FIRST_NAME, LAST_NAME
    INTO auxNom,auxApe
    FROM EMPLOYEES
    WHERE EMPLOYEE_ID = man;
    DBMS_OUTPUT.PUT_LINE(nom || ' ' || ape || ' tiene como jefe a ' || auxNom || ' ' || auxApe);
    CADENA_JEFES(man);
end if;

END CADENA_JEFES;
[2022-05-27 13:31:44] completed in 16 ms
```



```
Neena Kochhar tiene como jefe a Steven King
Steven King es el CEO
HR> begin
    --DBMS_OUTPUT.PUT_LINE(INVIERTE('daniel'));
    --LISTA_SEDES('United States of America');
    CADENA_JEFES(206);
end;
[2022-05-27 13:29:39] completed in 0 ms
William Gietz tiene como jefe a Shelley Higgins
Shelley Higgins tiene como jefe a Neena Kochhar
Neena Kochhar tiene como jefe a Steven King
Steven King es el CEO
```