

Proyecto RRHH

ÍNDICE:

CAMBIOS REQUERIDOS.....	2
A) Tabla EMPLE.....	2
B) Departamentos.....	2
C) Roles de empleados en departamento.....	2
E) Información adicional.....	3
ERD ACTUAL.....	4
ERD NUEVO.....	5
Dominio de valores:.....	6
MR.....	7
TABLA PL-SQL.....	8
LISTA DE DISPARADORES.....	9

CAMBIOS REQUERIDOS

A) Tabla EMPLE

- Renombrar a EMPLEADOS.
- Modificar datos:
 - Convertir salario y comisión de pesetas a euros. (Tasa de conversión: normalmente 1 euro = 166,386 pesetas).
 - Separar apellidos y nombre: deben quedar en columnas distintas.

B) Departamentos

- Añadir presupuesto anual (en miles de euros) para cada departamento.
- Registrar presupuestos históricos: ya que pueden cambiar cada año.

C) Roles de empleados en departamento

- Cada departamento puede tener:
 - 1 coordinador (solo uno).
 - Varios tesoreros (pueden ser más de uno).
- Restricciones:
 - Un empleado no puede tener más de un cargo en el mismo departamento.
 - Un empleado con cargo asignado no puede cambiar ni de cargo ni de departamento.

D) Reglas de integridad

- Todos los empleados deben:
 - Tener un salario (mínimo igual o superior al salario mínimo interprofesional de 2025).
 - Tener un jefe y un departamento asignados.
- Control presupuestario:
 - El gasto salarial total de un departamento no puede superar el 50% del presupuesto de ese año.

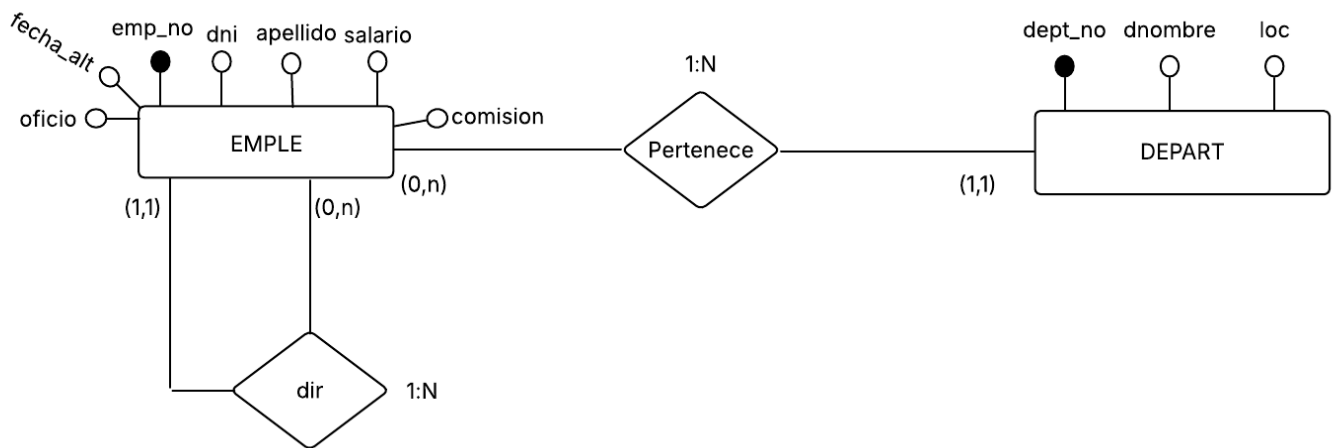
E) Información adicional

Salario bruto mensual, en 14 pagas.

Salario mínimo interprofesional en 2025? (1.184 €/mes en 2025 en España).

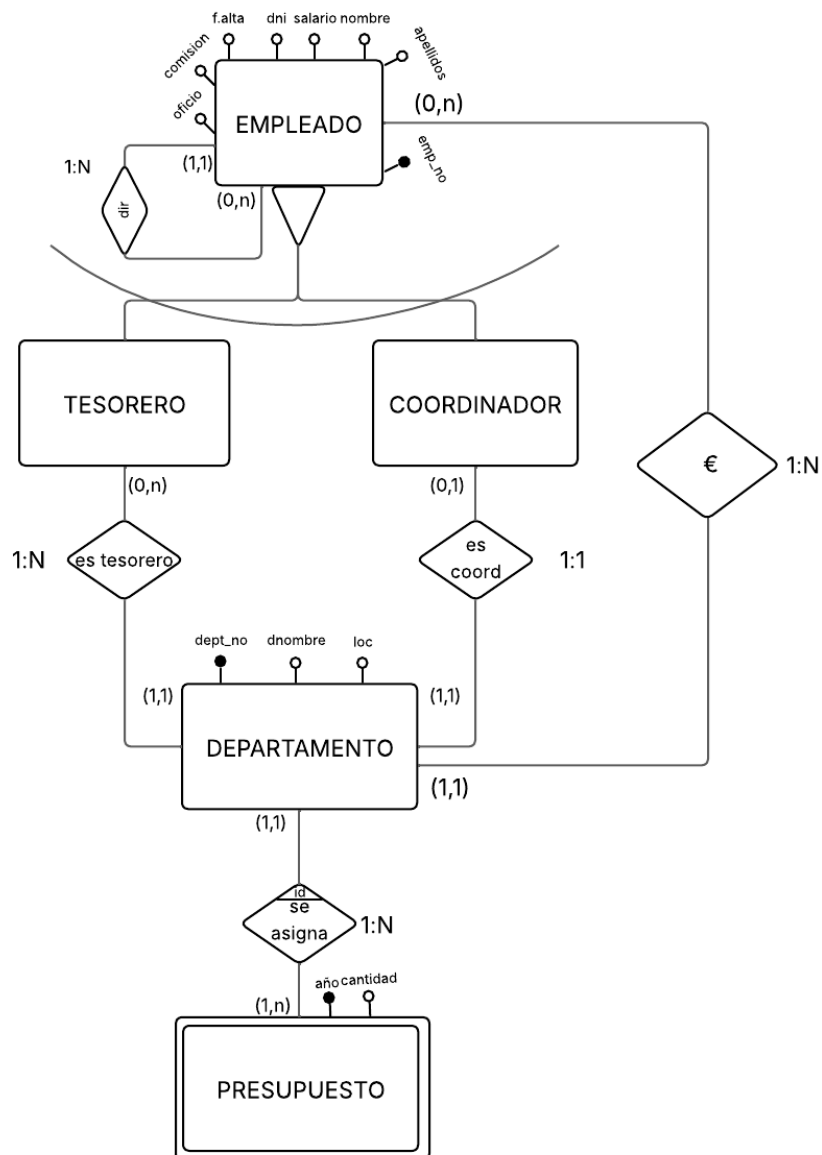
Tasa oficial de conversión pesetas/euros (lo habitual: 166,386 pesetas = 1 euro).

ERD ACTUAL



ERD NUEVO

- Garantizar que el depart en el que es Coordinador o tesorero es su depart
- Un empleado con cargo no puede cambiar de cargo ni de depart
- El total de los salarios de los emple no puede superar la mitad del presupuesto en ese depart
- Solo un empleado puede ser su propio jefe



Dominio de valores:

nombre x(30)
apellidos x(50)
emple_no x(3)
salario 9(8)
comision 9(5)
dni x(9)
fecha_alt date
dept_no x(2)
dnombre x(30)
loc x(30)
anio 9(4)
cantidad 9(12,2)

MR

DEPART(dept_no,emp_no_coord,dnombre,loc)

PK(dept_no)

FK(emp_no_coord)/EMPLEADOS

--emp_no_coord tiene que tener el mismo valor que emp_no en EMPLEADOS
--y tiene que haber las mismas tuplas--

PRESUPUESTOS(dept_no,año,cantidad)

PK(año,dept_no)

FK(dept_no)/DEPARTAMENTOS

--dept_no tiene que tener el mismo dominio que dept_no en DEPARTAMENTOS
--y tiene que haber las mismas tuplas--

<<"QUITANDO" SUBTIPOS>>

EMPLEADOS(cargo,emp_no,f.alta,dni,salario,nombre,apellidos,comision,oficio,emp_no_coord)

--cargo: (tesorero | coordinador) opcional--

PK(emp_no)

FK(emp_no_coord)/EMPLEADOS

--emp_no_coord tiene que tener el mismo dominio que emp_no en EMPLEADOS
--y tiene que haber las mismas tuplas--
--Garantizar que solo puede haber un cargo ya sea tesorero o coordinador.

DIR(emp_no_empleado,emp_no_jefe)

PK(emp_no_empleado)

FK(emp_no_empleado)/EMPLEADOS

--emp_no_empleado tiene que tener el mismo dominio que emp_no en EMPLEADOS--
--y tiene que haber las mismas tuplas--

DEPARTS(dept_no,dnombre,loc)

PK(dept_no)

--Garantizar que el departamento solo puede llegar a tener 1 coordinador.--

- Garantizar que el depart en el que es Coordinador o tesorero es su depart
- Un empleado con cargo no puede cambiar de cargo ni de depart
- El total de los salarios de los emple no puede superar la mitad del presupuesto en ese depart
- Solo un empleado puede ser su propio jefe
- Todos los empleados deben tener un salario mínimo igual o superior al salario mínimo interprofesional de 2025.

TABLA PL-SQL

CONTROL	TABLA	EVENTO	NECESITA VISTA
Maximo un coordinador	EMPLEADOS	Insert,update (TIPO)	EMPLEADOS
Un empleado con cargo no puede cambiar de cargo ni depart	EMPLEADOS	Update de (TIPO) Update de (DEPT_NO)	
Solo un jefe supremo	EMPLEADOS	insert,update de la tabla DIR (EMP_NO_JEFE)	EMPLEADOS
presupuesto >=doble del gasto salarial	EMPLEADOS	Insert update (SALARIO) update (DEPTNO)	EMPLEADOS
	PRESUPUESTOS	update(CANTIDAD)	
Todos los depart deben tener presupuesto	DEPART	INSERT	DEPART-PRESUPUESTO
Solo se puede modificar el presupuesto del año actual	PRESUPUESTOS	Update (CANTIDAD)	

LISTA DE DISPARADORES

Disparador 1 de la vista EMPLEADOS que controle:

- Solo un empleado que es su propio jefe.
 - Dep 10 7782 coordinador
 - Dep 20 nada
 - Dep 30 7499 coordinador 7521 tesorero
- Empleado no puede cambiar de cargo o dept.
- Máximo un coordinador por dept.

Disparador 2 de la vista DEPART-PRESUPUESTO que controle:

- Todos los los dept tienen que tener presupuesto.
- El presupuesto tiene que ser mayor o igual que el gasto salarial

Disparador 3 de la tabla PRESUPUESTO que controle:

- El presupuesto solo puede modificarse anualmente.

SCRIPTS

1. Crear las tablas nuevas:

```
CREATE TABLE PRESUPUESTOS(  
    DEPT_NO NUMBER(2) NOT NULL,  
    AÑO NUMBER(4) NOT NULL,  
    CANTIDAD NUMBER(12,2) NOT NULL, --10 digitos enteros y 2  
    decimales--  
    CONSTRAINT PK_PRESUPUESTOS PRIMARY KEY(DEPT_NO,  
    AÑO),  
    CONSTRAINT FK_PRESUPUESTOS_DEPART FOREIGN KEY  
    (DEPT_NO) REFERENCES DEPART  
);
```

```
CREATE TABLE EMPLEADOS(  
    EMP_NO NUMBER(4),  
    DNI CHAR(9) NOT NULL UNIQUE,  
    NOMBRE VARCHAR2(30) NOT NULL,  
    APELLIDOS VARCHAR2(60) NOT NULL,  
    OFICIO VARCHAR2(10),  
    FECHA_ALT DATE,  
    SALARIO NUMBER(10),  
    COMISION NUMBER(10),  
    DEPT_NO NUMBER(2) NOT NULL,  
    CONSTRAINT PK_EMPLEADOS PRIMARY KEY (EMP_NO),  
    CONSTRAINT FK_EMPLE_DEPARTS FOREIGN KEY (DEPT_NO)  
    REFERENCES DEPART  
);
```

```
CREATE TABLE DIR(  
    EMP_NO_EMPLEADO NUMBER(4),  
    EMP_NO_JEFE NUMBER(4),  
    CONSTRAINT PK_DIR PRIMARY KEY(EMP_NO_EMPLEADO),  
    CONSTRAINT FK_DIR_EMPLEADO FOREIGN  
    KEY(EMP_NO_EMPLEADO) REFERENCES EMPLEADOS);
```

2. Migrar los datos:

```
INSERT INTO EMPLEADOS(EMP_NO, DNI, NOMBRE, APELLIDOS,  
OFICIO, FECHA_ALT, SALARIO, COMISION, DEPT_NO)
```

```
SELECT
```

```
    EMP_NO,  
    DNI,  
    SUBSTR(APELLIDO, INSTR(APELLIDO, ',') + 1),  
    SUBSTR(APELLIDO, 1, INSTR(APELLIDO, ',') - 1),  
    OFICIO,  
    FECHA_ALT,  
    SALARIO / 166.386,  
    COMISION / 166.386,  
    DEPT_NO
```

```
FROM EMPL;
```

```
INSERT INTO DIR (EMP_NO_EMPLEADO,EMP_NO_JEFE)
```

```
SELECT
```

```
    EMP_NO,  
    DIR
```

```
FROM EMPL;
```

3. Crear vistas:

```
--VISTA EMPLEADOS--
```

```
CREATE OR REPLACE VIEW EMPLEADOS_EMPRESA AS
```

```
SELECT
```

```
    EMP_NO,  
    DNI,  
    NOMBRE,  
    APELLIDOS  
    OFICIO,  
    FECHA_ALT,  
    SALARIO,  
    COMISION,  
    DEPT_NO
```

FROM EMPLEADOS;

--VISTA DEPART-PRESUPUESTO Y MOSTRAR LOS DEPARTS
SINPRESUPUESTO - -

CREATE OR REPLACE VIEW DEPART_PRESUPUESTO AS
SELECT

d.DEPT_NO AS CODIGO_DEPARTAMENTO,
d.DNOMBRE AS NOMBRE_DEPARTAMENTO,
d.LOC AS LOCALIZACION,
p.AÑO AS ANIO_PRESUPUESTO,
p.CANTIDAD AS IMPORTE_PRESUPUESTO

FROM

DEPART d,
PRESUPUESTOS p

WHERE

d.DEPT_NO = p.DEPT_NO(+)

ORDER BY

d.DEPT_NO, p.AÑO DESC;

SELECT

e.DEPT_NO,
d.DNOMBRE,
SUM(e.SALARIO * 14) AS GASTO_SALARIAL,
SUM(e.SALARIO * 14) * 2 AS PRESUPUESTO_MINIMO

FROM

EMPLEADOS e

WHERE e.DEPT_NO=d.DEPT_NO(+)

GROUP BY

e.DEPT_NO, d.DNOMBRE

ORDER BY

e.DEPT_NO;

4. Migrar datos a presupuestos con procedimiento:

```
SET SERVEROUTPUT ON;
DECLARE
    anio NUMBER := TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY'));
    factor NUMBER := 2; -- 2x el gasto salarial
    presupuesto NUMBER;
BEGIN
    FOR depto IN (
        SELECT
            e.DEPT_NO,
            d.DNOMBRE,
            NVL(SUM(e.SALARIO * 14), 0) AS gasto_salarial
        FROM
            EMPLEADOS e,
            DEPART d
        WHERE
            e.DEPT_NO = d.DEPT_NO(+)
        GROUP BY
            e.DEPT_NO, d.DNOMBRE
    ) LOOP
        -- Calcular presupuesto minimo
        presupuesto := (depto.gasto_salarial * factor);

        -- Insertar en PRESUPUESTOS
        INSERT INTO PRESUPUESTOS (DEPT_NO, AÑO, CANTIDAD)
        VALUES (depto.DEPT_NO, anio, presupuesto);

        DBMS_OUTPUT.PUT_LINE(
            RPAD('Depto ' || depto.DEPT_NO || ' ' || depto.DNOMBRE, 30)
            ||
            RPAD('Gasto anual: ' || depto.gasto_salarial, 25) ||
            'Presupuesto: ' || presupuesto
        );
    END LOOP;
```

```
COMMIT;
DBMS_OUTPUT.PUT_LINE('Presupuestos asignados a todos los
departamentos');
EXCEPTION
WHEN OTHERS THEN
ROLLBACK;
DBMS_OUTPUT.PUT_LINE('Error: ' || SUBSTR(SQLERRM, 1,
200));
END;
/
```

5. Crear triggers:

--Disparador 1 de la vista EMPLEADOS que controle:

--Solo un empleado que es su propio jefe.

--Empleado no puede cambiar de cargo o dept.

--Máximo un coordinador por dept.

```
CREATE OR REPLACE TRIGGER AUDITA_EMPLEADOS  
BEFORE INSERT OR UPDATE OF OFICIO, DEPT_NO ON  
EMPLEADOS
```

```
FOR EACH ROW
```

```
DECLARE
```

```
    es_jefe_supremo NUMBER;
```

```
    coordinador_existente NUMBER;
```

```
    tiene_cargo_especial NUMBER;
```

```
    es_su_propio_jefe NUMBER;
```

```
BEGIN
```

```
    -- 1. Control: Solo un empleado puede ser su propio jefe (jefe  
supremo)
```

```
    -- Verificar si este empleado es su propio jefe
```

```
BEGIN
```

```
    SELECT 1 INTO es_su_propio_jefe
```

```
    FROM DIR
```

```
    WHERE EMP_NO_EMPLEADO = :NEW.EMP_NO
```

```
    AND EMP_NO_JEFE = :NEW.EMP_NO;
```

```
    -- Si es su propio jefe, verificar que no haya otro jefe supremo
```

```
    SELECT COUNT(*) INTO es_jefe_supremo
```

```
    FROM DIR
```

```
    WHERE EMP_NO_EMPLEADO = EMP_NO_JEFE
```

```
    AND EMP_NO_EMPLEADO != :NEW.EMP_NO;
```

```
    IF es_jefe_supremo > 0 THEN
```



```
        RAISE_APPLICATION_ERROR(-20001, 'Ya existe un jefe  
supremo en la organización');
```

```
    END IF;
```

```
EXCEPTION
```

```
    WHEN NO_DATA_FOUND THEN
```

```
        es_su_propio_jefe :=0; -- No es su propio jefe
```

```
    END;
```

```
-- 2. Control: Máximo un coordinador por departamento
```

```
IF (:NEW.OFICIO = 'COORDINADOR') AND
```

```
    (INSERTING OR (UPDATING AND (:OLD.OFICIO !=
```

```
'COORDINADOR' OR :OLD.DEPT_NO != :NEW.DEPT_NO))) THEN
```

```
    SELECT COUNT(*) INTO coordinador_existente
```

```
    FROM EMPLEADOS_EMPRESA
```

```
    WHERE DEPT_NO = :NEW.DEPT_NO
```

```
    AND OFICIO = 'COORDINADOR'
```

```
    AND EMP_NO != NVL(:OLD.EMP_NO, 0);
```

```
IF coordinador_existente > 0 THEN
```

```
    RAISE_APPLICATION_ERROR(-20002, 'Ya existe un  
coordinador en el departamento ' || :NEW.DEPT_NO);
```

```
    END IF;
```

```
END IF;
```

```
-- 3. Control: Empleados con cargo no pueden cambiar de cargo ni  
departamento
```

```
IF UPDATING THEN
```

```
    -- Verificar si tenía un cargo especial antes
```

```
    SELECT COUNT(*) INTO tiene_cargo_especial
```

```
    FROM EMPLEADOS_EMPRESA
```

```
    WHERE EMP_NO = :OLD.EMP_NO
```

```
    AND OFICIO IN ('COORDINADOR', 'TESORERO');
```

```
IF tiene_cargo_especial > 0 THEN
```

```
    -- No puede cambiar de cargo
```

```
    IF :NEW.OFICIO != :OLD.OFICIO THEN
```

```

        RAISE_APPLICATION_ERROR(-20003, 'Empleados con
cargo (' || :OLD.OFICIO || ') no pueden cambiar de cargo');
    END IF;

```

```

        -- No puede cambiar de departamento
        IF :NEW.DEPT_NO != :OLD.DEPT_NO THEN
            RAISE_APPLICATION_ERROR(-20004, 'Empleados con
cargo (' || :OLD.OFICIO || ') no pueden cambiar de departamento');
        END IF;
    END IF;
END IF;
END;
/

```

--Disparador 2 de la vista DEPART-PRESUPUESTO que controle:

--Todos los los dept tienen que tener presupuesto.

--El presupuesto tiene que ser mayor o igual que el gasto salarial

```

CREATE OR REPLACE TRIGGER AUDITA_PRESUPUESTOS
BEFORE INSERT OR UPDATE OR DELETE ON PRESUPUESTOS
FOR EACH ROW
DECLARE
    gasto_salarial NUMBER;
    dept_sin_presupuesto NUMBER;
    presupuesto_minimo NUMBER;
BEGIN
    -- Para INSERT/UPDATE: Controlar relación presupuesto-gasto
    salarial
    IF INSERTING OR UPDATING THEN
        -- 1. Calcular el gasto salarial del departamento (14 pagas)
        SELECT NVL(SUM(e.SALARIO * 14), 0)
        INTO total_salarios
        FROM EMPLEADOS e
        WHERE e.DEPT_NO = :NEW.DEPT_NO;
    
```

-- 2. Calcular el presupuesto mínimo requerido (doble del gasto salarial)

presupuesto_minimo := total_salarios * 14 * 2

-- 3. Validar que el presupuesto asignado cumple con el mínimo

IF :NEW.CANTIDAD < presupuesto_minimo THEN

RAISE_APPLICATION_ERROR(-20020,

'Presupuesto insuficiente para el departamento ' ||

:NEW.DEPT_NO || CHR(10) ||

'Gasto salarial anual: ' || total_salarios || CHR(10) ||

'Presupuesto mínimo requerido (2x): ' || presupuesto_minimo ||
CHR(10) ||

'Presupuesto asignado: ' || :NEW.CANTIDAD);

END IF;

END IF;

-- Después de cualquier cambio: Verificar que todos los dept tengan presupuesto

IF INSERTING OR UPDATING OR DELETING THEN

FOR dept_rec IN (

SELECT d.DEPT_NO

FROM DEPART d

WHERE NOT EXISTS (

SELECT 1

FROM PRESUPUESTOS p

WHERE p.DEPT_NO = d.DEPT_NO

AND p.AÑO = TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY'))

)

) LOOP

RAISE_APPLICATION_ERROR(-20011,

'El departamento ' || dept_rec.DEPT_NO ||

' no tiene presupuesto para el año actual');

END LOOP;

END IF;

END;

/

--Disparador 3 de la tabla PRESUPUESTO que controle:

--El presupuesto solo puede modificarse anualmente.

CREATE OR REPLACE TRIGGER

AUDITA_CAMBIO_PRESUPUESTO

BEFORE UPDATE OF CANTIDAD ON PRESUPUESTOS

FOR EACH ROW

DECLARE

anio_actual NUMBER;

BEGIN

-- Obtener el año actual

anio_actual := TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY'));

-- Verificar si el presupuesto que se intenta modificar es del año actual

IF :OLD.AÑO != anio_actual THEN

RAISE_APPLICATION_ERROR(-20030,

'Solo se puede modificar el presupuesto del año actual (' ||
anio_actual || '). Intenta modificar el año ' || :OLD.AÑO);

END IF;

END;

/