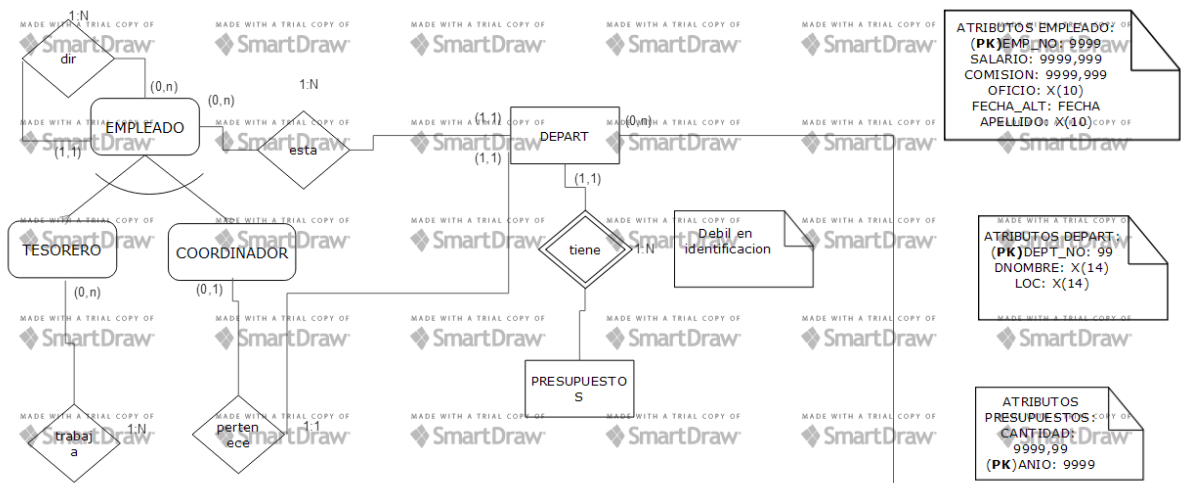


ERD



MR

//MODELO RELACIONAL:

(*) DECIDÍ ELIMINAR LA JERARQUÍA DEBIDO A QUE AL ELIMINAR EL SUBTIPO TENDRÍAMOS QUE TRABAJAR LA EXISTENCIA DE UN SOLO COORDINADOR POR UN DEPARTAMENTO, SIN EMBARGO AL HACERLO DE ESTA MANERA TENDRÍAMOS SOLO UN CASO ESPECIAL, EL COORDINADOR, YA QUE TANTO EL TESORERO COMO UN EMPLEADO CORRIENTE COMPARTEN LA MISMA RELACION CON EL DEPARTAMENTO PODEMOS HACERLO SIMPLEMENTE ASÍ, QUEDANDO UNA RELACIÓN 1:1 CON LA TABLA EMPLEADO. PERO ASÍ TENDRIAMOS QUE TRABAJAR CON TABLAS REDUNDADAS, SIENDO INNECESARIO.

EMPLEADO(EMP_NO,DEPT_NO,DIR,SALARIO,COMISION,OFICIO,FECHA_ALT,APELLIDO,TIPO_EMPLE)

PK(EMP_NO)

FK(DEPT_NO) NN / DEPART

FK(DIR) NN / EMPLEADO

TIPO_EMPLE X(12) ["TESORERO", "COORDINADOR", NULL]

(*) POR CADA DE EMPLEADO, SOLO PUEDE HABER UNA TUPLA EN DEPART

DEPART(DEPT_NO,DNOMBRE,LOC)

PK(DEPT_NO) NN

PRESUPUESTO(ANIO,DEPT_NO,CANTIDAD)

PK(ANIO,DEPT_NO)

FK(DEPT_NO) NN / DEPART

(*)ELEGÍ ESTA OPCION YA QUE DE ESTA MANERA TANTO LA RELACION POR IDENTIFICACIÓN COMO LA CARDINALIDAD 1:N PODAMOS CONTRALARLA POR PROGRAMACIÓN SABRIENDO QUE POR CADA TUPLA EN DEPART DEBE HABER UNA EN PRESUPUESTOS

SCRIPT

```
CREATE TABLE PRESUPUESTOS (  
    ANIO NUMBER(4) NOT NULL,  
    DEPT_NO    NUMBER(2) NOT NULL,  
    CANTIDAD   NUMBER(6,2) NOT NULL,  
    CONSTRAINT PK_PRESUPUESTOS PRIMARY KEY (ANIO,DEPT_NO) ,  
    CONSTRAINT FK_DEPT_NO_PRESUPUESTOS FOREIGN KEY (DEPT_NO)  
        REFERENCES DEPART ON DELETE CASCADE ,  
    CONSTRAINT CK_CANTIDAD CHECK(CANTIDAD>0) ,  
    CONSTRAINT CK_ANIO CHECK(ANIO>=1582)
```

```
);
```

```
--CREANDO LA NUEVA TABLA
```

```
--CAMBIOS DOMINIO DE DATOS
```

```
-----
```

```
CREATE TABLE EMPLEADO(  
    EMP_NO    NUMBER(4) NOT NULL,  
    APELLIDO   VARCHAR2(10) ,  
    OFICIO     VARCHAR2(10) ,  
    DIR        NUMBER(4) ,  
    FECHA_ALT  DATE ,  
    SALARIO    NUMBER(10) ,  
    COMISION   NUMBER(10) ,  
    DEPT_NO    NUMBER(2)
```

```
);
```

```
ALTER TABLE EMPLEADO
```

```
    MODIFY SALARIO NUMBER(7,3);
```

```
ALTER TABLE EMPLEADO
```

```
    MODIFY COMISION NUMBER(7,3);
```

```
ALTER TABLE EMPLEADO
```

```
    ADD TIPO_EMPLE VARCHAR2(12);
```

--MODIFICAR LA TABLA ADECUADA

--CONSTRAINTS

--DE EMPLEADO

ALTER TABLE EMPLEADO

ADD CONSTRAINT PK_EMPLEADO PRIMARY KEY(EMP_NO);

ALTER TABLE EMPLEADO

ADD CONSTRAINT FK_DEPT_NO_EMPLEADO FOREIGN KEY (DEPT_NO) REFERENCES
DEPART;

ALTER TABLE EMPLEADO

ADD CONSTRAINT FK_DIR_EMPLEADO FOREIGN KEY (DIR) REFERENCES EMPLEADO;

ALTER TABLE EMPLEADO

ADD CONSTRAINT CK_TIPO_EMPL_EMPLEADO CHECK (TIPO_EMPL='TESORERO'
OR TIPO_EMPL='COORDINADOR' OR TIPO_EMPL=NULL);

ALTER TABLE EMPLEADO

MODIFY SALARIO NOT NULL;

ALTER TABLE EMPLEADO

MODIFY DIR NOT NULL;

ALTER TABLE EMPLEADO

MODIFY FECHA_ALT NOT NULL;

ALTER TABLE EMPLEADO

ADD CONSTRAINT CK_COMISION CHECK (COMISION>=0);

ALTER TABLE EMPLEADO

MODIFY DEPT_NO NOT NULL;

--RESTRICCION DEL SALARIO MINIMO

ALTER TABLE EMPLEADO

ADD CONSTRAINT CK_SALARIO CHECK (TRUNC(SALARIO,2)>=625.05);

--DATOS INSERTAR

--UPDATES EMPLEADO Y TIPO DE EMPLEADO

INSERT INTO EMPLEADO (EMP_NO,APELLIDO,
OFICIO,DIR,FECHA_ALT,SALARIO,COMISION,DEPT_NO)

SELECT EMP_NO,APELLIDO,
OFICIO,DIR,FECHA_ALT,SALARIO/166.386,COMISION/166.386,DEPT_NO FROM
EMPLE

ORDER BY DEPT_NO;

UPDATE EMPLEADO

SET TIPO_EMPL='TESORERO'

WHERE EMP_NO=7521;

UPDATE EMPLEADO

SET TIPO_EMPL='COORDINADOR'

WHERE EMP_NO=7782;

```

UPDATE EMPLEADO
  SET TIPO_EMPLE='COORDINADOR'
  WHERE EMP_NO=7499;

--TRIGGERS
--VISTAS NECESARIAS

CREATE OR REPLACE VIEW EMPLEADOS AS SELECT * FROM EMPLEADO ORDER BY
EMP_NO;

CREATE OR REPLACE VIEW DEPARTAMENTOS AS SELECT
DEPART.DEPT_NO,DNOMBRE,LOC,ANIO,CANTIDAD FROM DEPART, PRESUPUESTOS
  WHERE DEPART.DEPT_NO=PRESUPUESTOS.DEPT_NO;

CREATE OR REPLACE VIEW TODOS_LOS_PRESUPUESTOS AS SELECT * FROM
PRESUPUESTOS;

CREATE OR REPLACE TRIGGER EMPL_DEPART INSTEAD OF INSERT OR DELETE OR
UPDATE ON EMPLEADOS
FOR EACH ROW
DECLARE
jefesSupremos NUMBER(1);
coordinadores NUMBER(1);
salarioDepartamento NUMBER(10,2);
cantidadDepartamentoEsteAnio NUMBER(6,2);
BEGIN
IF INSERTING THEN
  SELECT COUNT(*) INTO jefesSupremos FROM EMPLEADO
    WHERE EMP_NO=DIR;
  IF jefesSupremos>=1 AND :new.DIR = :new.EMP_NO THEN
    RAISE_APPLICATION_ERROR(-20002,'YA EXISTE UN JEFE SUPREMO');
  END IF;
  SELECT COUNT(*) INTO coordinadores FROM EMPLEADO
    WHERE TIPO_EMPLE='COORDINADOR' AND DEPT_NO=:new.DEPT_NO;
  IF coordinadores=1 AND :new.TIPO_EMPLE='COORDINADOR' THEN
    RAISE_APPLICATION_ERROR(-20003,'YA EXISTE UN COORDINADOR EN
ESE DEPARTAMENTO');
  END IF;
  INSERT INTO EMPLEADO
VALUES (:new.EMP_NO,:new.APELLIDO,:new.OFICIO,:new.DIR,:new.FECHA_ALT,
:new.SALARIO,:new.COMISION,:new.DEPT_NO,:NEW.TIPO_EMPLE);
END IF;
IF UPDATING THEN
IF UPDATING('DEPT_NO') THEN
  UPDATE EMPLEADO

```

```

        SET DEPT_NO=:new.DEPT_NO
        WHERE EMP_NO=:old.EMP_NO;
    END IF;

    SELECT CANTIDAD INTO cantidadDepartamentoEsteAnio FROM
    PRESUPUESTOS WHERE DEPT_NO=:new.DEPT_NO AND
    ANIO=TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY'));

    SELECT SUM(SALARIO+COMISION)+:new.SALARIO+:new.COMISION INTO
    salarioDepartamento FROM EMPLEADO
    WHERE DEPT_NO=:new.DEPT_NO;

    IF salarioDepartamento*2>=cantidadDepartamentoEsteAnio THEN
        RAISE_APPLICATION_ERROR(-20004, 'SUPERADO LA MITAD DEL
    PRESUPUESTO DE ESTE AÑO PARA ESTE DEPARTAMENTO');
    END IF;

    IF UPDATING('SALARIO') THEN
        UPDATE EMPLEADO
        SET SALARIO=:NEW.SALARIO
        WHERE EMP_NO=:old.EMP_NO;
    END IF;

    IF UPDATING('COMISION') THEN
        UPDATE EMPLEADO
        SET COMISION=:NEW.COMISION
        WHERE EMP_NO=:old.EMP_NO;
    END IF;

    IF UPDATING('TIPO_EMPL') THEN
        SELECT COUNT(*) INTO coordinadores FROM EMPLEADO
        WHERE TIPO_EMPL='COORDINADOR' AND DEPT_NO=:new.DEPT_NO;
        IF coordinadores=1 AND :new.TIPO_EMPL='COORDINADOR' THEN
            RAISE_APPLICATION_ERROR(-20003, 'YA EXISTE UN COORDINADOR EN
    ESE DEPARTAMENTO');
        END IF;
        UPDATE EMPLEADO
        SET TIPO_EMPL=:NEW.TIPO_EMPL
        WHERE EMP_NO=:old.EMP_NO;
    END IF;

    IF UPDATING('DIR') THEN
        SELECT COUNT(*) INTO jefesSupremos FROM EMPLEADO
        WHERE EMP_NO=DIR;
        IF jefesSupremos>=1 AND :new.DIR = :new.EMP_NO THEN
            RAISE_APPLICATION_ERROR(-20002, 'YA EXISTE UN JEFE SUPREMO');
        END IF;
    
```

```

        UPDATE EMPLEADO
        SET DIR=:NEW.DIR
        WHERE EMP_NO=:old.EMP_NO;
    END IF;
    IF UPDATING('OFICIO') THEN
        UPDATE EMPLEADO
    SET OFICIO=:NEW.OFICIO
    WHERE EMP_NO=:old.EMP_NO;
    END IF;
    IF UPDATING('APELLIDO') THEN
    UPDATE EMPLEADO
    SET APELLIDO=:NEW.APELLIDO
    WHERE EMP_NO=:old.EMP_NO;
    END IF;
    IF UPDATING('EMP_NO') THEN
        UPDATE EMPLEADO
        SET EMP_NO=:new.EMP_NO
        WHERE EMP_NO=:old.EMP_NO;
    END IF;
END IF;
IF DELETING THEN
    DBMS_OUTPUT.PUT_LINE('NO PUEDE BORRAR DESDE ESTA TABLA');
END IF;
END;

/
CREATE OR REPLACE TRIGGER PRESUPUESTO
INSTEAD OF INSERT OR DELETE OR UPDATE ON TODOS_LOS_PRESUPUESTOS
FOR EACH ROW
DECLARE
salarioDepartamento NUMBER(10,2);
cantidadDepartamentoEsteAnio NUMBER(6,2);
numEmpleadosDepart NUMBER(4);
BEGIN
    IF UPDATING('ANIO') THEN
        RAISE_APPLICATION_ERROR(-20010,'LOS CAMPOS DE AÑO NO PUEDEN
MODIFICARSE');
    END IF;
    IF UPDATING('CANTIDAD') AND
:NEW.ANIO=TO_NUMBER(TO_CHAR(SYSDATE,'yyyy')) THEN

```

```

SELECT CANTIDAD INTO cantidadDepartamentoEsteAnio FROM
PRESUPUESTOS WHERE DEPT_NO=:new.DEPT_NO AND
ANIO=TO_NUMBER(TO_CHAR(SYSDATE,'yyyy'));
        SELECT SUM(SALARIO+COMISION) INTO
salarioDepartamento FROM EMPLEADO
        WHERE DEPT_NO=:new.DEPT_NO;
        IF salarioDepartamento*2>=cantidadDepartamentoEsteAnio THEN
                RAISE_APPLICATION_ERROR(-20004,'SUPERADO LA MITAD DEL
PRESUPUESTO DE ESTE AÑO PARA ESTE DEPARTAMENTO');
        END IF;
        UPDATE PRESUPUESTOS
        SET CANTIDAD=:NEW.CANTIDAD
        WHERE ANIO=:OLD.ANIO AND DEPT_NO=:OLD.DEPT_NO;
ELSE
        RAISE_APPLICATION_ERROR(-21000,'NO PUEDE MODIFICAR PRESUPUESTOS
DE OTROS AÑOS');
        END IF;
        IF DELETING THEN
                SELECT COUNT(*) INTO numEmpleadosDepart FROM EMPLEADOS
                WHERE DEPT_NO=:old.DEPT_NO;
                IF numEmpleadosDepart>0 THEN
                        RAISE_APPLICATION_ERROR(-21001,'HAY EMPLEADOS EN ESE
DEPARTAMENTO O HA INTENTADO BORRAR UN PRESUPUESTOS DE ESTE AÑO
');
                END IF;
                DELETE FROM PRESUPUESTOS
                WHERE DEPT_NO=:OLD.DEPT_NO AND ANIO=:OLD.ANIO;
        END IF;
        END;
/

```

```

CREATE OR REPLACE TRIGGER DEPART_PRESUPUEST
INSTEAD OF INSERT OR DELETE OR UPDATE ON DEPARTAMENTOS
FOR EACH ROW
DECLARE
        salarioDepartamento NUMBER(10,2);
        cantidadDepartamentoEsteAnio NUMBER(6,2);
BEGIN

```

```

        SELECT CANTIDAD INTO cantidadDepartamentoEsteAnio FROM
PRESUPUESTOS WHERE DEPT_NO=:new.DEPT_NO AND
ANIO=TO_NUMBER(TO_CHAR(SYSDATE,'yyyy'));
        SELECT SUM(SALARIO+COMISION) INTO
                salarioDepartamento FROM EMPLEADO
                WHERE DEPT_NO=:new.DEPT_NO;
        IF INSERTING AND :new.ANIO=TO_NUMBER(TO_CHAR(SYSDATE,'yyyy'))
THEN
        INSERT INTO DEPART
        VALUES (:new.DEPT_NO,:new.DNOMBRE,:new.LOC);
        INSERT INTO PRESUPUESTO VALUES
        (:new.ANIO,:new.DEPT_NO,:new.CANTIDAD);
        ELSE
        RAISE_APPLICATION_ERROR(-21000,'INSERTE RESPECTO AL AÑO
ACTUAL');
        END IF;
        IF DELETING THEN
        DBMS_OUTPUT.PUT_LINE('NO PUEDE ELIMINAR UN DEPARTAMENTO DESDE
ESTA VISTA');
        END IF;
END;

/

```

--TERMINAR CON LA TABLA ANTERIOR CON LOS DATOS CAMBIADOS

DROP TABLE EMPLE CASCADE CONSTRAINTS;

--LAS PUEDES INSERTAR PRESUPUESTOS DIRECTAMENTE EN SU TABLA, PERO SI QUIERES CREAR UNO NUEVO, DEBES USAR LA VISTA DEPARTAMENTOS, Y MODIFICAR LA VISTA TODOS_LOS_PRESUPUESTOS PARA CAMBIAR O BORRAR DATOS DE ESA TABLA