

-----Creación de Tablas-----

```
CREATE TABLE COMPRAR_PEDIDO (  
    ID_PEDIDO          NUMBER NOT NULL,  
    ESTADO  VARCHAR2(30) NOT NULL,  
    FECHA    DATE,  
    CANTIDAD FLOAT(4) NOT NULL,  
    CONSTRAINT COMPRAR_PEDIDO_PK PRIMARY KEY(ID_PEDIDO)  
);
```

```
CREATE TABLE FIJAR_NOMINA(  
    ID_NOMINA NUMBER NOT NULL,  
    SUELDO FLOAT(5),  
    FECHA_CONTRATO DATE,  
    CONSTRAINT FIJAR_NOMINA_PK PRIMARY KEY (ID_NOMINA)  
);
```

```
CREATE TABLE TRABAJADOR_COBRA_NOMINA(  
    ID_TRABAJADOR NUMBER NOT NULL,  
    ID_NOMINA NUMBER NOT NULL,  
    NOMBRE VARCHAR2(20),  
    TLF      NUMBER(9),  
    DNI VARCHAR2(9),  
    EMAIL VARCHAR2(20),  
    SUELDO FLOAT(5),  
    FECHA_CONTRATO DATE,  
    CARGA_TRABAJO NUMBER,  
    CONSTRAINT TRABAJADOR_COBRA_NOMINA_PK PRIMARY KEY (ID_TRABAJADOR,  
ID_NOMINA),  
    CONSTRAINT TRABAJADOR_COBRA_NOMINA_FK1 FOREIGN KEY (SUELDO)  
REFERENCES FIJAR_NOMINA(SUELDO),  
    CONSTRAINT TRABAJADOR_COBRA_NOMINA_FK2 FOREIGN KEY (FECHA_CONTRATO)  
REFERENCES FIJAR_NOMINA(FECHA_CONTRATO)  
);
```

```
CREATE TABLE REGISTRAR_CATEGORIA(  
    ID_CATEGORIA NUMBER NOT NULL,  
    NOMBRE VARCHAR2(20),  
    CONSTRAINT REGISTRAR_CATEGORIA_PK PRIMARY KEY (ID_CATEGORIA)  
);
```

```

CREATE TABLE SISTEMA_CONTRATA_TRABAJADOR (
    ID_TRABAJADOR NUMBER NOT NULL,
    NOMBRE VARCHAR2(20),
    TLF NUMBER(9),
    DNI VARCHAR2(9),
    EMAIL VARCHAR2(20),
    CONSTRAINT CTRABAJADOR_PK PRIMARY KEY (ID_TRABAJADOR),
    CONSTRAINT CTRABAJADOR_FK1 FOREIGN KEY (ID_TRABAJADOR) REFERENCES
TRABAJADOR_COBRA_NOMINA(ID_TRABAJADOR),
    CONSTRAINT CTRABAJADOR_FK2 FOREIGN KEY (NOMBRE) REFERENCES
TRABAJADOR_COBRA_NOMINA(NOMBRE),
    CONSTRAINT CTRABAJADOR_FK3 FOREIGN KEY (TLF) REFERENCES
TRABAJADOR_COBRA_NOMINA(TLF),
    CONSTRAINT CTRABAJADOR_FK4 FOREIGN KEY (DNI) REFERENCES
TRABAJADOR_COBRA_NOMINA(DNI),
    CONSTRAINT CTRABAJADOR_FK5 FOREIGN KEY (EMAIL) REFERENCES
TRABAJADOR_COBRA_NOMINA(EMAIL)
);

```

```

CREATE TABLE BLOQUEA (
    ID_TRABAJADOR NUMBER NOT NULL,
    CONSTRAINT BLOQUEA_PK PRIMARY KEY (ID_TRABAJADOR),
    CONSTRAINT BLOQUEA_FK1 FOREIGN KEY (ID_TRABAJADOR) REFERENCES
TRABAJADOR_COBRA_NOMINA(ID_TRABAJADOR)
);

```

```

CREATE TABLE GERENTE (
    ID_TRABAJADOR NUMBER NOT NULL,
    NOMBRE VARCHAR2(20),
    TLF NUMBER(9),
    DNI VARCHAR2(9),
    EMAIL VARCHAR2(20),
    DESPACHO NUMBER,
    CONSTRAINT GERENTE_PK PRIMARY KEY (ID_TRABAJADOR),
    CONSTRAINT GERENTE_FK1 FOREIGN KEY (ID_TRABAJADOR) REFERENCES
TRABAJADOR_COBRA_NOMINA(ID_TRABAJADOR),
    CONSTRAINT GERENTE_FK2 FOREIGN KEY (NOMBRE) REFERENCES
TRABAJADOR_COBRA_NOMINA(NOMBRE),
    CONSTRAINT GERENTE_FK3 FOREIGN KEY (TLF) REFERENCES
TRABAJADOR_COBRA_NOMINA(TLF),
    CONSTRAINT GERENTE_FK4 FOREIGN KEY (DNI) REFERENCES
TRABAJADOR_COBRA_NOMINA(DNI),
    CONSTRAINT GERENTE_FK5 FOREIGN KEY (EMAIL) REFERENCES
TRABAJADOR_COBRA_NOMINA(EMAIL)
);

```

```

CREATE TABLE SISTEMA_CREA_PROYECTO(
    ID_PROYECTO NUMBER NOT NULL,
    ID_GERENTE NUMBER NOT NULL,
    FECHA_INICIO DATE,
    DURACION NUMBER,
    NOMBRE_CLIENTE VARCHAR2(20),
    DESCRIPCION VARCHAR2(50),
    ESTADO VARCHAR2(10),
    CONSTRAINT SISTEMA_CREA_PROYECTO_PK PRIMARY KEY (ID_PROYECTO,
NOMBRE_CLIENTE),
    CONSTRAINT SISTEMA_CREA_PROYECTO_FK1 FOREIGN KEY (ID_GERENTE)
REFERENCES TRABAJADOR_COBRA_NOMINA(ID_TRABAJADOR)
);

```

```

CREATE TABLE ASIGNADO(
    ID_PROYECTO NUMBER NOT NULL,
    ID_TRABAJADOR NUMBER NOT NULL,
    CONSTRAINT ASIGNADO_PK PRIMARY KEY (ID_PROYECTO),
    CONSTRAINT ASIGNADO_FK1 FOREIGN KEY (ID_PROYECTO) REFERENCES
SISTEMA_CREA_PROYECTO(ID_PROYECTO),
    CONSTRAINT ASIGNADO_FK2 FOREIGN KEY (ID_TRABAJADOR) REFERENCES
TRABAJADOR_COBRA_NOMINA(ID_TRABAJADOR)
);

```

```

CREATE TABLE EMPLEADO_CATEGORIA(
    ID_TRABAJADOR NUMBER NOT NULL,
    ID_CATEGORIA NUMBER NOT NULL,
    NOMBRE VARCHAR2(20),
    TLF NUMBER(9),
    DNI VARCHAR2(9),
    EMAIL VARCHAR2(20),
    NOMBRE_CATEGORIA VARCHAR2(20),
    CONSTRAINT EMPLEADO_CATEGORIA_PK PRIMARY KEY (ID_TRABAJADOR,
ID_CATEGORIA),
    CONSTRAINT EMPLEADO_CATEGORIA_FK1 FOREIGN KEY (ID_TRABAJADOR)
REFERENCES TRABAJADOR_COBRA_NOMINA(ID_TRABAJADOR),
    CONSTRAINT EMPLEADO_CATEGORIA_FK2 FOREIGN KEY (NOMBRE) REFERENCES
TRABAJADOR_COBRA_NOMINA(NOMBRE),
    CONSTRAINT EMPLEADO_CATEGORIA_FK3 FOREIGN KEY (TLF) REFERENCES
TRABAJADOR_COBRA_NOMINA(TLF),
    CONSTRAINT EMPLEADO_CATEGORIA_FK4 FOREIGN KEY (DNI) REFERENCES
TRABAJADOR_COBRA_NOMINA(DNI),
    CONSTRAINT EMPLEADO_CATEGORIA_FK5 FOREIGN KEY (EMAIL) REFERENCES
TRABAJADOR_COBRA_NOMINA(EMAIL),
    CONSTRAINT EMPLEADO_CATEGORIA_FK6 FOREIGN KEY (ID_CATEGORIA)
REFERENCES REGISTRAR_CATEGORIA(ID_CATEGORIA),
    CONSTRAINT EMPLEADO_CATEGORIA_FK7 FOREIGN KEY (NOMBRE_CATEGORIA)
REFERENCES REGISTRAR_CATEGORIA(NOMBRE)
);

```

```

CREATE TABLE PROVEEDOR_ALMACENA_MATERIA(
    ID_PROVEEDOR NUMBER NOT NULL,
    ID_MATERIA NUMBER NOT NULL,
    NOMBRE_PROVEEDOR VARCHAR2(20),
    TLF NUMBER(9),
    NIF VARCHAR2(9),
    EMAIL VARCHAR2(20),
    NOMBRE_MATERIA VARCHAR2(20),
    PRECIO FLOAT(5),
    CONSTRAINT PROVEEDOR_ALMACENA_MATERIA_PK PRIMARY KEY (ID_PROVEEDOR,
ID_MATERIA)
);

```

```

CREATE TABLE REGISTRAR_NECESIDADES(
    ID_PROYECTO NUMBER NOT NULL,
    CATEGORIA NUMBER NOT NULL,
    ID_MATERIA NUMBER NOT NULL,
    CONSTRAINT REGISTRAR_NECESIDADES_PK PRIMARY KEY (ID_PROYECTO),
    CONSTRAINT REGISTRAR_NECESIDADES_FK1 FOREIGN KEY (CATEGORIA)
REFERENCES REGISTRAR_CATEGORIA(ID_CATEGORIA),
    CONSTRAINT REGISTRAR_NECESIDADES_FK2 FOREIGN KEY (ID_MATERIA)
REFERENCES PROVEEDOR_ALMACENA_MATERIA(ID_MATERIA)
);

```

-----Creación de Procedimientos-----

**--1 -Crear Proyecto**

```
CREATE OR REPLACE PROCEDURE CREAM_PROYECTO(NOMBRE_CLIENTE VARCHAR2,
ID_GERENTE NUMBER, DESCRIPCION VARCHAR2, CATEGORIA NUMBER, FECHA_INICIO
DATE, DURACION NUMBER) IS
```

```
    STATE VARCHAR2;
    CODPRO NUMBER;
    CODPRONew NUMBER;
```

```
BEGIN
```

```
    SELECT ID_PROYECTO INTO CODPRO FROM SISTEMA_CREA_PROYECTO WHERE
ROWNUM = 1 ORDER BY DESC;
```

```
    STATE := 'Iniciado';
    INSERT INTO SISTEMA_CREA_PROYECTO VALUES(CODPRONew, ID_GERENTE,
FECHA_INICIO, DURACION, NOMBRE_CLIENTE, DESCRIPCION, STATE);
```

```
END;
```

**--2 -Finalizar Proyecto**

```
CREATE OR REPLACE PROCEDURE FINALIZAR_PROYECTO(ID_PRO NUMBER) IS
```

```
BEGIN
```

```
    UPDATE SISTEMA_CREA_PROYECTO SET ESTADO = 'Finalizado' WHERE
ID_PROYECTO = ID_PRO;
END;
```

**--3 -Registrar Área Categoría**

```
CREATE OR REPLACE PROCEDURE REGISTRAR_AREA_CATEGORIA(ID_CATEGORIA NUMBER,
NOMBRE_CATEGORIA VARCHAR2) IS
```

```
BEGIN
```

```
    INSERT INTO REGISTRAR_CATEGORIA VALUES (ID_CATEGORIA,
NOMBRE_CATEGORIA);
END;
```

**--4 -Asignar Carga a Empleado**

```
CREATE OR REPLACE PROCEDURE ASIGNAR_CARGA_EMPLEADO(ID_TRAB NUMBER,
PORCENTAJE NUMBER) IS
```

```
BEGIN
```

```
    UPDATE TRABAJADOR_COBRA_NOMINA SET CARGA_TRABAJO = PORCENTAJE WHERE
ID_TRABAJADOR = ID_TRAB;
END;
```

**--5 -Consultar Carga de un Trabajador**

```
CREATE OR REPLACE PROCEDURE CONSULTAR_CARGA(ID_TRAB) IS
CNT NUMBER;

BEGIN
    SELECT COUNT(*) INTO CNT FROM TRABAJADOR_COBRA_NOMINA WHERE
ID_TRABAJADOR = ID_TRAB;
END;
```

**-----Creación de Disparadores-----**

**--1 -El porcentaje de carga total de un empleado no puede ser mayor al 100%**

```
CREATE OR REPLACE TRIGGER CARGA_TOTAL
    BEFORE INSERT OR UPDATE OF CARGA_TRABAJO ON TRABAJADOR_COBRA_NOMINA
FOR EACH ROW

    DECLARE
        CNT NUMBER;

BEGIN
    SELECT COUNT(*) INTO CNT FROM TRABAJADOR_COBRA_NOMINA;

    IF(CNT + :NEW.CARGA_TRABAJO > 100) THEN
        RAISE_APPLICATION_ERROR(-20001, 'El porcentaje de carga total
de un empleado no puede ser mayor al 100%');
    END IF;

END;
```

**--2 -Para registrar a un empleado en un proyecto, debe pertenecer a su misma categoría**

```
CREATE OR REPLACE TRIGGER CATEMPLEADO
    BEFORE INSERT OR UPDATE OF ID_TRABAJADOR ON ASIGNADO FOR EACH ROW

DECLARE
    CATNEC VARCHAR2;
    CATEMP VARCHAR2;

BEGIN
    SELECT CATEGORIA INTO CATNEC FROM REGISTRAR_NECESIDADES
    WHERE ID_PROYECTO = (SELECT ID_PROYECTO FROM SISTEMA_CREA_PROYECTO
                                WHERE ID_PROYECTO = (SELECT
ID_PROYECTO FROM ASIGNADO
WHERE ID_TRABAJADOR = :NEW.ID_TRABAJADOR)));

    IF(CATNEC != CATEMP) THEN
        RAISE_APPLICATION_ERROR(-20002, 'La categoría requerida por
el proyecto y la categoría del empleado no coinciden');
    END IF;

END;
```

**--3 -La fecha de inicio de un proyecto debe ser superior a la fecha actual**

```
CREATE OR REPLACE TRIGGER FECHA_CORRECTA
    BEFORE INTO OR UPDATE OF FECHA_INICIO ON SISTEMA_CREA_PROYECTO FOR
    EACH ROW
    DECLARE
        ACTUAL DATE;

BEGIN
    SELECT SYSDATE INTO ACTUAL FROM DUAL

    IF(ACTUAL != :NEW.FECHA_INICIO) THEN
        RAISE_APPLICATION_ERROR(-20003, 'La fecha de inicio de un
proyecto debe ser superior a la fecha actual');
    END IF;
END;
```