

UT4 – Diseño Físico de Bases de Datos

Organizaciones No Gubernamentales (ONG)

En los siguientes apartados váis a encontrar el proceso completo de diseño de una base de datos, cubriendo todos los pasos que hemos visto: descripción del sistema, diseño conceptual, diseño lógico y terminando con unos scripts SQL que nos generarán desde cero la estructura de la base de datos en el SDBD Oracle.

1.- Descripción de Requisitos del Sistema

La coordinadora nacional de Organizaciones No Gubernamentales (ONG's) desea mantener una base de datos de asociaciones de este tipo que existen en nuestro país. Para ello necesita almacenar la información sobre cada asociación, los socios que las componen, los proyectos que realizan y los trabajadores de las mismas.

De las asociaciones se desea almacenar su CIF, denominación, dirección y provincia, su tipo (ecologista, integración, desarrollo, ...), así como si está declarada de utilidad pública por el Ministerio del Interior.

Cada asociación está formada por socios de los que se precisa conocer su DNI, nombre, dirección, provincia, fecha de alta en la asociación, la cuota mensual con la que colaboran y la aportación anual que realizan (que se obtendrá multiplicando la cuota mensual por los meses del año)

Los trabajadores de estas organizaciones pueden ser de dos tipos: asalariados y voluntarios.

Los asalariados son trabajadores que cobran un sueldo y ocupan cierto cargo en la asociación. Se desea almacenar la cantidad que éstos pagan a la seguridad social y el tanto por ciento del IRPF que se les descuenta.

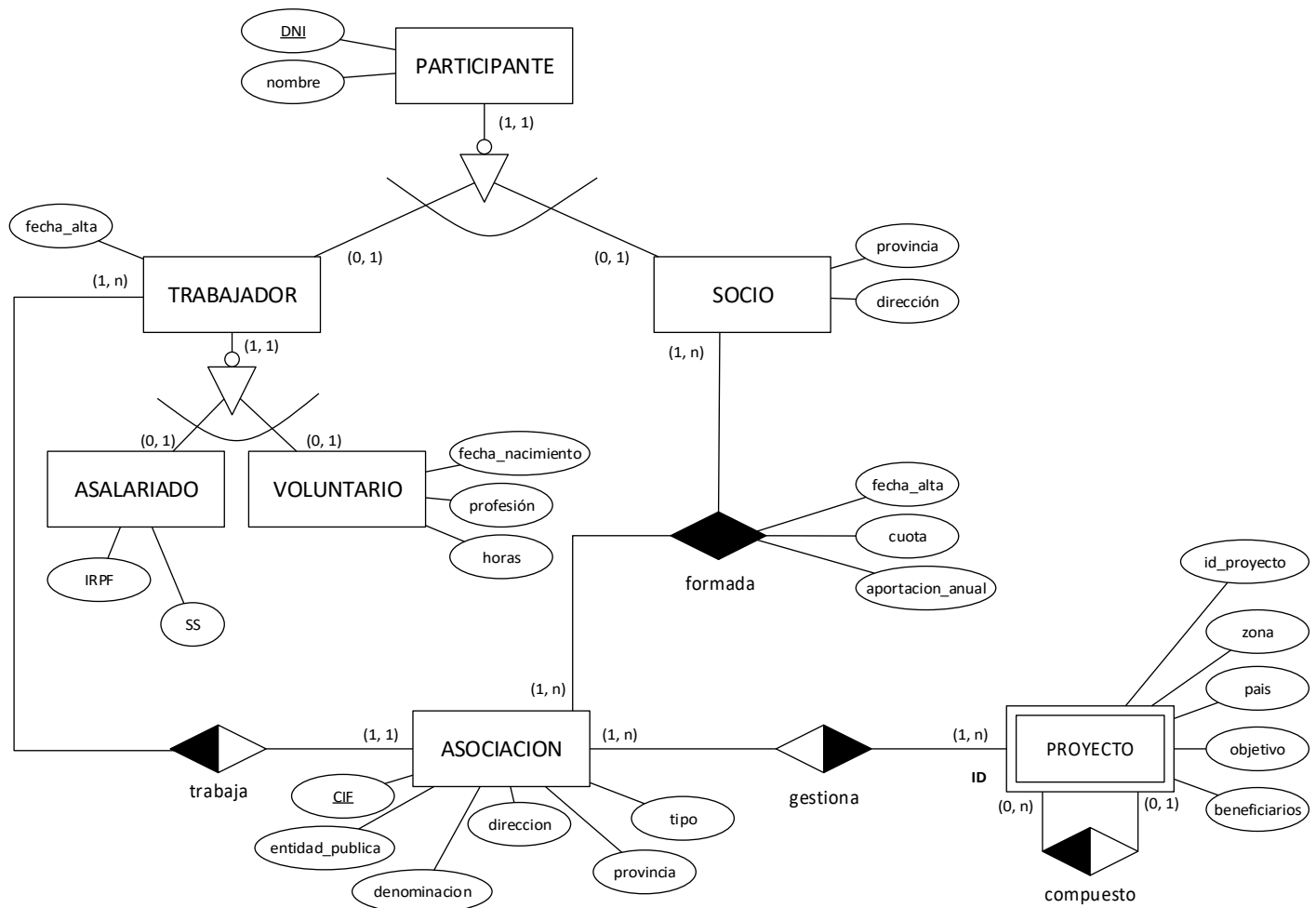
Los voluntarios trabajan en la organización desinteresadamente, siendo preciso conocer su edad, profesión y las horas que dedican a la asociación a efectos de cálculo de estadísticas.

Cada trabajador se identifica por su DNI, tiene un nombre y una fecha de ingreso.

Un socio no puede ser trabajador de la asociación.

Las asociaciones llevan a cabo proyectos a los que están asignados sus trabajadores. De cada proyecto se desea almacenar su número de identificación dentro de la asociación, en qué país se lleva a cabo y en qué zona de éste, así como el objetivo que persigue y el número de beneficiarios a los que afecta. Un proyecto se compone a su vez de subproyectos (que tienen entidad de proyectos).

2.- Diseño Conceptual: Esquema Entidad Relación



3.- Diseño Lógico: Modelo Relacional

PARTICIPANTE(DNI, nombre)
PK: DNI

SOCIO(DNI, provincia, direccion)
PK: DNI
FK: DNI -> PARTICIPANTE

TRABAJADOR(DNI, fecha_alta, CIF_asociacion)
PK: DNI
FK: DNI -> PARTICIPANTE
FK: cif_asociacion -> ASOCIACION

ASALARIADO(DNI, IRPF, SS)
PK: DNI
FK: DNI -> TRABAJADOR

VOLUNTARIO(DNI, fecha_nacimiento, profesion, horas)
PK: DNI
FK: DNI -> TRABAJADOR

ASOCIACION(CIF, denominacion, provincia, direccion, entidad_publica, tipo)
PK: CIF

FORMADA(CIF_asociacion, DNI_socio, fecha_alta, cuota, aportacion_anual)
PK: (CIF_asociacion, DNI_socio)
FK: CIF_asociacion -> ASOCIACION
FK: DNI_socio -> SOCIO

PROYECTO(CIF_asociacion, id_proyecto, zona, pais, objetivo, beneficiarios, id_padre)
PK: (CIF_asociacion, id_proyecto)
FK: CIF_asociacion -> ASOCIACION
FK: (CIF_asociacion, id_padre) -> PROYECTO

4.- Diseño Físico: Oracle SQL

Partiendo de una instalación completa de Oracle Database y la aplicación SQL Developer, el proceso que seguiremos para crear la base de datos consistiría en:

- Crear un tablespace que será nuestro espacio de trabajo, utilizando el usuario `system`
- Crear un usuario para trabajar en el tablespace creado
- Conectarnos a Oracle Database con el nuevo usuario en SQL Developer
- Crear todas las tablas en base al diseño lógico realizado
- Cargar las tablas con algunos datos de prueba

Las sentencias SQL para llevar a cabo estos pasos se han agrupado en tres archivos que será con los que trabajaremos en SQL Developer. Se deberán ejecutar en el orden en que se indica.

En primer lugar nos conectaremos a Oracle Database con el usuario DBA (`system` en Oracle), creando un nuevo tablespace (utilizad vuestro directorio de `oradata`) y nuestro usuario de trabajo.

```
-----  
-- crea_workspace_ong.sql  
-- creamos el tablespace y el usuario que vamos a utilizar en la base de datos ONG  
-- ejecutar con usuario system  
  
-- creamos el tablespace donde la BD almacenará sus datos  
CREATE SMALLFILE TABLESPACE "ongBD"  
DATAFILE 'C:\app\jmanogil\product\18.0.0\oradata\XE\ongbd.dbf' SIZE 100M  
LOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;  
  
-- comprobamos que se ha creado correctamente  
SELECT tablespace_name, status FROM dba_tablespaces;  
  
-- creamos el usuario que trabajará contra esa nueva BD  
ALTER SESSION SET "_ORACLE_SCRIPT"=true;  
CREATE USER ong IDENTIFIED BY ong DEFAULT TABLESPACE "ongBD" QUOTA UNLIMITED ON "ongBD"  
TEMPORARY TABLESPACE "TEMP";  
  
-- comprobamos que se ha creado  
SELECT * FROM all_users ORDER BY created DESC;  
  
-- le asignamos los roles necesarios  
-- en principio no haría falta CREATE VIEW ni CREATE SYNONIM, pero se los damos  
GRANT CONNECT, RESOURCE, CREATE VIEW, CREATE SYNONYM TO ong;  
  
-----
```

A continuación, nos conectaremos con el nuevo usuario y crearemos todas las tablas

```
-----  
-- crea_bd_ong.sql  
-- Una vez creado el tablespace y el usuario ong, creamos las tablas de esta BD  
-- utilizando el usuario creado para ello, esto es, conectándonos con una conexión que  
-- emplee el usuario ong  
  
-- BORRADO TABLAS  
-- eliminamos todas la tablas por si estuvieran ya creadas  
-- el orden de borrado también está influido por las foreign keys  
-- de esas tablas, debemos empezar borrando las tablas con FK a otras  
-- dejando para el final aquellas que están referenciadas por las FK  
-- de las primeras  
  
-- Oracle no tiene la instrucción DROP TABLE IF EXISTS  
DROP TABLE asalariado;  
DROP TABLE voluntario;  
DROP TABLE trabajador;  
DROP TABLE formada;  
DROP TABLE socio;  
DROP TABLE participante;  
DROP TABLE proyecto;  
DROP TABLE asociacion;  
  
-- CREACION TABLAS  
-- el orden de creación de las tablas también tendrá en cuenta las FK, lógicamente  
-- no podemos crear una tabla que tenga una FK hacia una tabla que todavía no hemos creado  
-- deberíamos hacer un proceso en orden inverso al de borrado  
  
-- tabla PARTICIPANTE  
CREATE TABLE participante (  
DNI VARCHAR2(10),  
nombre NVARCHAR2(50) NOT NULL,  
CONSTRAINT pk_participante PRIMARY KEY (DNI));  
  
-- tabla SOCIO  
CREATE TABLE socio (  
DNI VARCHAR2(10),  
provincia NVARCHAR2(50) NOT NULL,  
direccion NVARCHAR2(100) NOT NULL,  
CONSTRAINT pk_socio PRIMARY KEY (DNI),  
CONSTRAINT fk_socio_participante FOREIGN KEY(DNI)  
REFERENCES participante(DNI) ON DELETE CASCADE);
```

```
-- tabla ASOCIACION
CREATE TABLE asociacion (
CIF VARCHAR2(10),
denominacion NVARCHAR2(50) NOT NULL,
provincia NVARCHAR2(50),
direccion NVARCHAR2(100),
tipo NVARCHAR2(50) NOT NULL,
entidad_publica NUMBER(1) NOT NULL CHECK (entidad_publica in (0,1)),
CONSTRAINT pk_asociacion PRIMARY KEY (CIF));

-- tabla TRABAJADOR
CREATE TABLE trabajador (
DNI VARCHAR2(10),
fecha_alta DATE NOT NULL,
CIF_asociacion VARCHAR2(10),
CONSTRAINT pk_trabajador PRIMARY KEY (DNI),
CONSTRAINT fk_trabajador_participante FOREIGN KEY(DNI)
REFERENCES participante(DNI) ON DELETE CASCADE,
CONSTRAINT fk_trabajador_asociacion FOREIGN KEY(CIF_asociacion)
REFERENCES asociacion(CIF) ON DELETE CASCADE);

-- tabla ASALARIADO
CREATE TABLE asalariado (
DNI VARCHAR2(10),
SS VARCHAR2(15),
IRPF NUMBER(4, 2),
CONSTRAINT pk_asalariado PRIMARY KEY (DNI),
CONSTRAINT fk_asalariado_trabajador FOREIGN KEY(DNI)
REFERENCES trabajador(DNI) ON DELETE CASCADE,
CONSTRAINT ck_irpf CHECK(IRPF > 0 AND IRPF < 100));

-- tabla VOLUNTARIO
CREATE TABLE voluntario (
DNI VARCHAR2(10),
fecha_nacimiento DATE NOT NULL,
profesion NVARCHAR2(50),
horas NUMBER(4, 1),
CONSTRAINT pk_voluntario PRIMARY KEY (DNI),
CONSTRAINT fk_voluntario_trabajador FOREIGN KEY(DNI)
REFERENCES trabajador(DNI) ON DELETE CASCADE);
```

```
-- tabla FORMADA
CREATE TABLE formada (
CIF_asociacion VARCHAR2(10),
DNI_socio VARCHAR2(10),
fecha_alta DATE NOT NULL,
cuota NUMBER(10, 2) NOT NULL,
aportacion_anual NUMBER(10, 2) NOT NULL,
CONSTRAINT pk_formada PRIMARY KEY (CIF_asociacion, DNI_socio),
CONSTRAINT fk_formada_asociacion FOREIGN KEY(CIF_asociacion) REFERENCES asociacion(CIF),
CONSTRAINT fk_formada_socio FOREIGN KEY(DNI_socio) REFERENCES socio(DNI));

-- tabla PROYECTO
CREATE TABLE proyecto (
id_proyecto NUMBER(6),
CIF_asociacion VARCHAR2(10),
id_padre NUMBER(6),
pais NVARCHAR2(50),
zona NVARCHAR2(50),
objetivo NVARCHAR2(2000),
beneficiarios NUMBER(10),
CONSTRAINT pk_proyecto PRIMARY KEY(CIF_asociacion, id_proyecto),
CONSTRAINT fk_proyecto_asociacion FOREIGN KEY(CIF_asociacion)
REFERENCES asociacion(CIF) ON DELETE CASCADE,
CONSTRAINT fk_proyecto_proyecto
FOREIGN KEY(CIF_asociacion, id_padre)
REFERENCES proyecto(CIF_asociacion, id_proyecto) ON DELETE CASCADE);
-----
```

Por último, vamos a introducir algunos datos en las tablas, también usando la conexión con el usuario creado para trabajar con esta base de datos

```
-----
-- carga_bd_ong.sql
-- Una vez creadas todas las tablas de la BD ong, cargamos algunas filas en ellas
-- utilizando el usuario creado para ello, esto es, conectándonos con una conexión que
-- emplee el usuario ong

-- Hay que tener en cuenta que las foreign keys nos obligan a seguir un orden
-- en la carga de las tablas

-- para trabajar con fechas formato ISO YYYY-MM-DD,
-- en lugar del formato por defecto de Oracle
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD';
```

```
-- VACIADO TABLAS

-- Podríamos usar la instrucción TRUNCATE para vaciar las tablas pero Oracle
-- nos obliga a desactivar las FK, por lo que usaremos DELETE
-- hay que tener en cuenta que los borrados en cascada de las FK nos
-- ahorra mucho trabajo
DELETE FROM formada;
DELETE FROM participante; -- vacía también socio, trabajador, asalariado y voluntario
DELETE FROM asociacion; -- vacía también proyecto

-- CARGA TABLAS

-- tabla PARTICIPANTE
INSERT INTO participante VALUES ('1111A', 'PEDRO LÓPEZ');
INSERT INTO participante VALUES ('2222B', 'ANA GARCÍA');
INSERT INTO participante VALUES ('3333C', 'MARÍA CUESTA');
INSERT INTO participante VALUES ('4444D', 'LUÍS SEGURA');

-- tabla SOCIO
INSERT INTO socio VALUES ('1111A', 'ALICANTE', 'CALLE MAYOR 25, 2A');
INSERT INTO socio VALUES ('2222B', 'MURCIA', 'CALLE SEGURA 52, BAJO A');

-- tabla ASOCIACION
INSERT INTO asociacion
VALUES ('1234567G', 'SOLIDARIDAD SIN FRONTERAS', NULL, NULL, 'INTEGRACION', 1);

INSERT INTO asociacion
VALUES ('7654321T', 'AMIGOS DE LOS GATOS',
'ALICANTE', 'CARRETERA ALICANTE KM 56', 'ECOLOGISTA', 1);

-- tabla TRABAJADOR
INSERT INTO trabajador VALUES ('3333C', '2015-01-20', '1234567G');
INSERT INTO trabajador VALUES ('4444D', '2016-12-06', '7654321T');

-- tabla ASALARIADO
INSERT INTO asalariado VALUES ('3333C', '123/456', 15.25);

-- tabla VOLUNTARIO
INSERT INTO voluntario VALUES ('4444D', '1995-05-20', 'INFORMATICO', 30.5);

-- tabla FORMADA
INSERT INTO formada VALUES ('1234567G', '1111A', '2017-06-10', 10, 120);
INSERT INTO formada VALUES ('7654321T', '2222B', '2017-09-01', 20, 240);
```

```
-- tabla PROYECTO
INSERT INTO proyecto VALUES (56,
'1234567G', NULL, 'ESPAÑA', 'LEVANTE', 'Menores inmigrantes', 50);

INSERT INTO proyecto VALUES (108, '1234567G', 56, 'ESPAÑA', 'LEVANTE', 'Minusválidos', 10);

INSERT INTO proyecto VALUES (120, '1234567G', 56, 'ESPAÑA', 'LEVANTE', 'Autistas', 15);
```
