

PREGUNTAS	1	2	3	4	NOTA
PUNTOS	2	2	3	3	
PUNTUACIÓN					

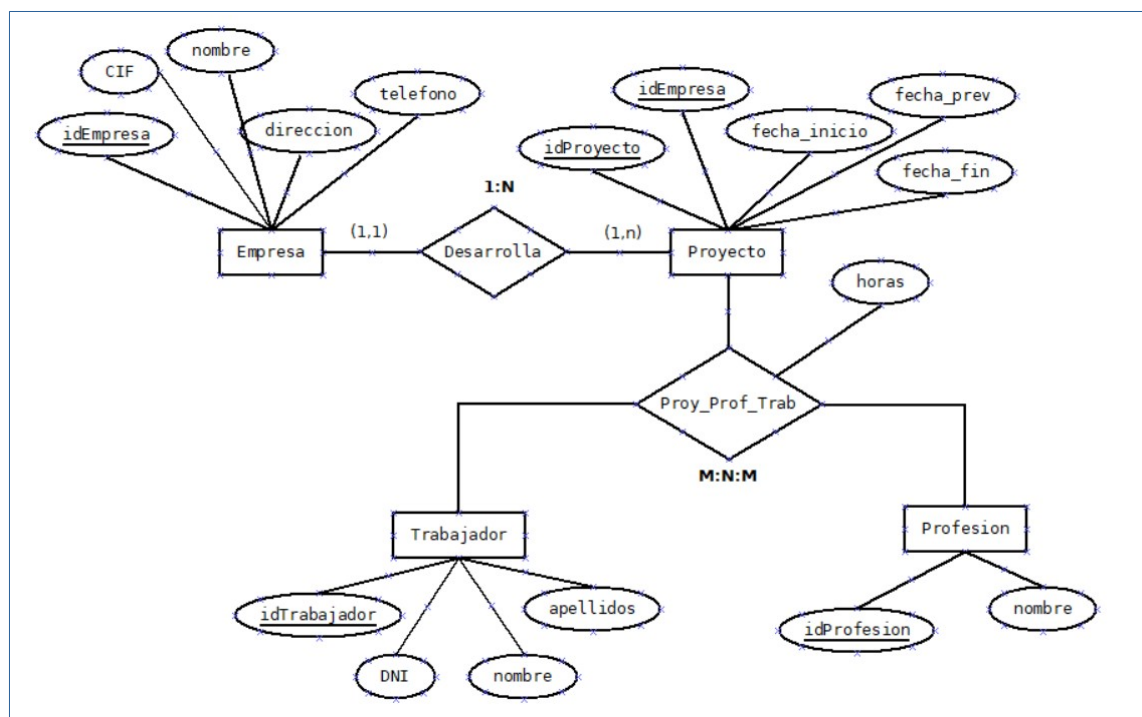
Tarea 2

Ejercicio 1: Empresa de software:

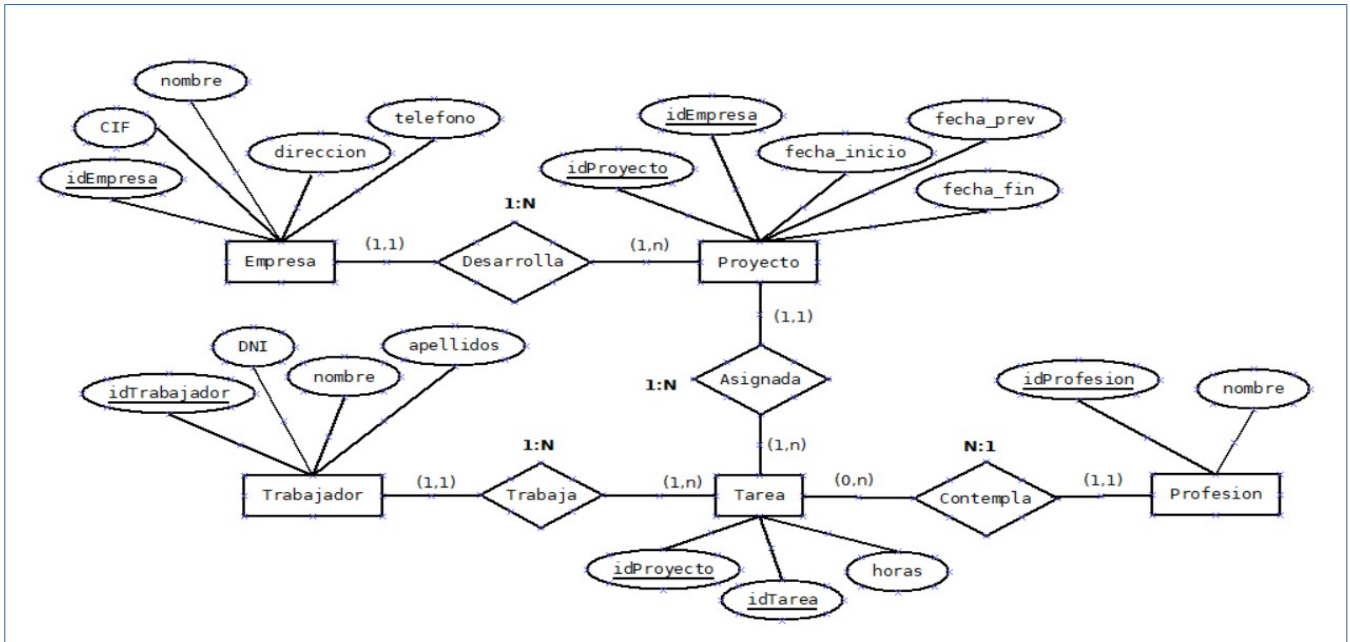
Realizar un esquema entidad/relación (**notación Chen**) utilizando una herramienta similar a **Dia Diagram Editor** que permita modelar el sistema de información de una empresa de software atendiendo las siguientes premisas:

- La empresa crea proyectos para otras empresas. De dichas empresas se almacena el CIF, nombre, dirección y teléfono así como un código interno de empresa.
- Los proyectos se inician en una determinada fecha y finalizan en otra. Además al planificarle se almacena la fecha prevista de finalización (que puede no coincidir con la finalización real)
- Los proyectos los realizan varios trabajadores, cada uno de ellos desempeña una determinada profesión en el proyecto (analista, jefe de proyecto, programador,...), dicha profesión tiene un código de profesión. En el mismo proyecto puede haber varios analistas, programadores,...
- Todos los trabajadores tienen un código de trabajador, un dni, un nombre y apellidos. Su profesión puede cambiar según el proyecto: en uno puede ser jefe y en otro un programador
- Se anota las horas que ha trabajado cada trabajador en cada proyecto.
- Puede haber varios proyectos que comiencen el mismo día.
- A todas las empresas les hemos realizado al menos un proyecto
- Todos los trabajadores han participado en algún proyecto
- En la base de datos, la profesión "administrador de diseño" no la ha desempeñado todavía ningún trabajador o trabajadora

MODELO CONCEPTUAL



OTRA FORMA: TRANSFORMACIÓN DE RELACIÓN TERNARIA



Ejercicio 2: Realizar el ejercicio anterior utilizando la notación “pata de gallo” especificando las claves primarias y ajenas (foráneas) de cada relación (entidad).

- Utilice el editor de SQL Developer Data Modeler

PASO DEL MODELO CONCEPTUAL AL ESQUEMA RELACIONAL (NO GRÁFICO)

Empresa (idEmpresa, cif, nombre, direccion, telefono)

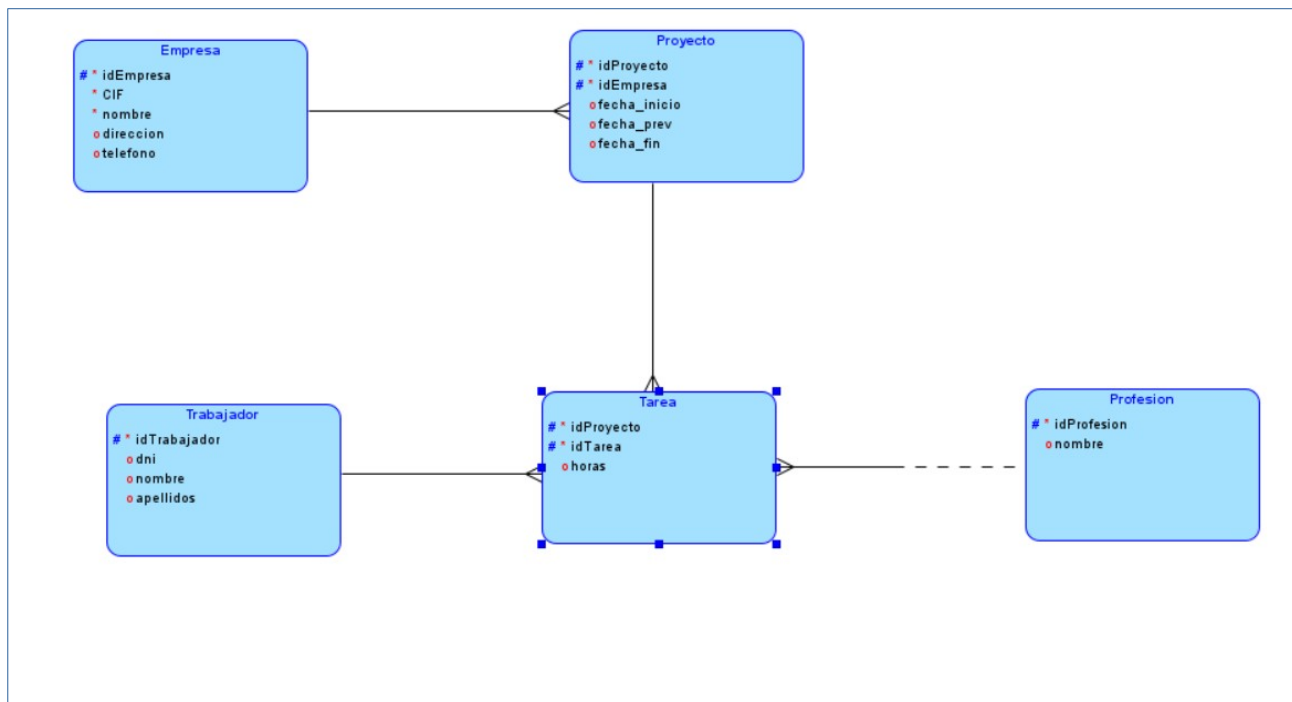
Proyecto (idEmpresa, idProyecto, fecha_inicio, fecha_prev, fecha_fin)

Tarea (idProyecto, idProfesion, idTrabajador, horas,)

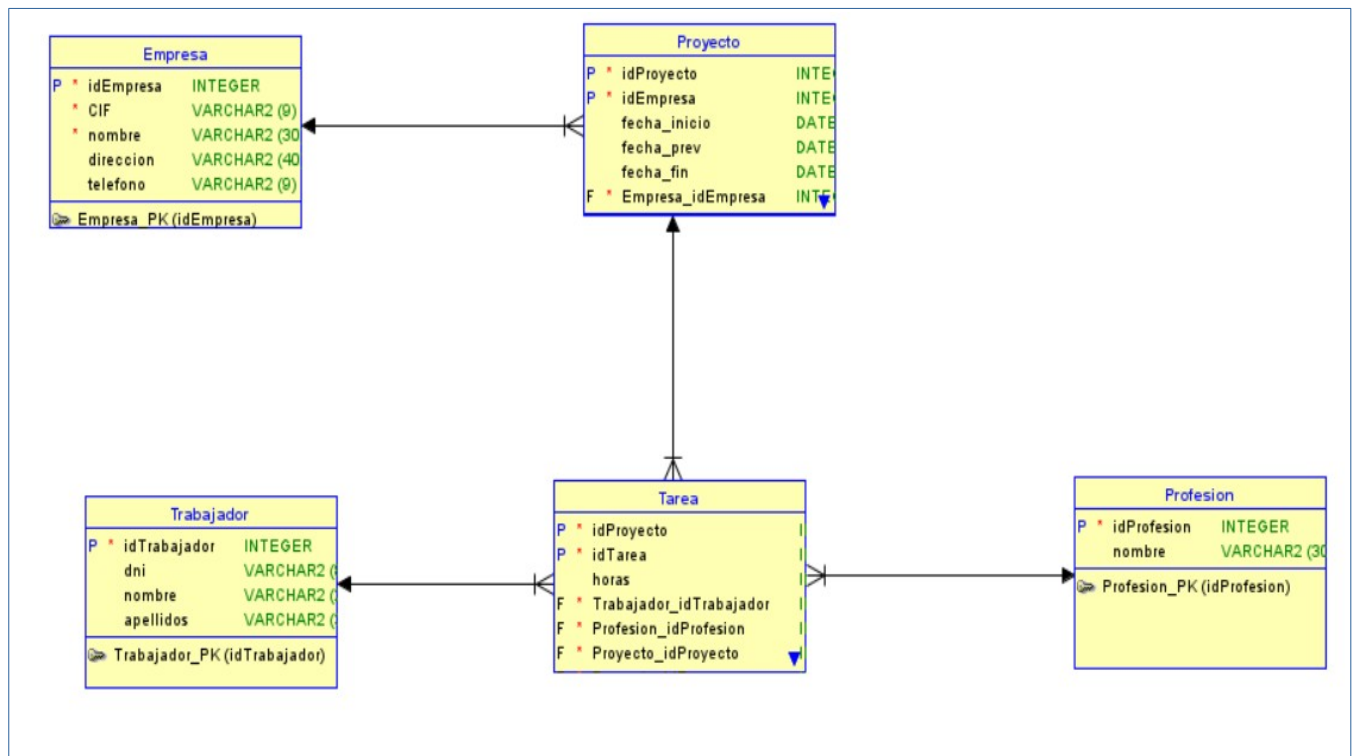
Trabajador (idTrabajador, dni, nombre, apellidos)

Profesion (idProfesion, nombre)

Modelo lógico en Data Modeler



Modelo relacional en Data Modeler

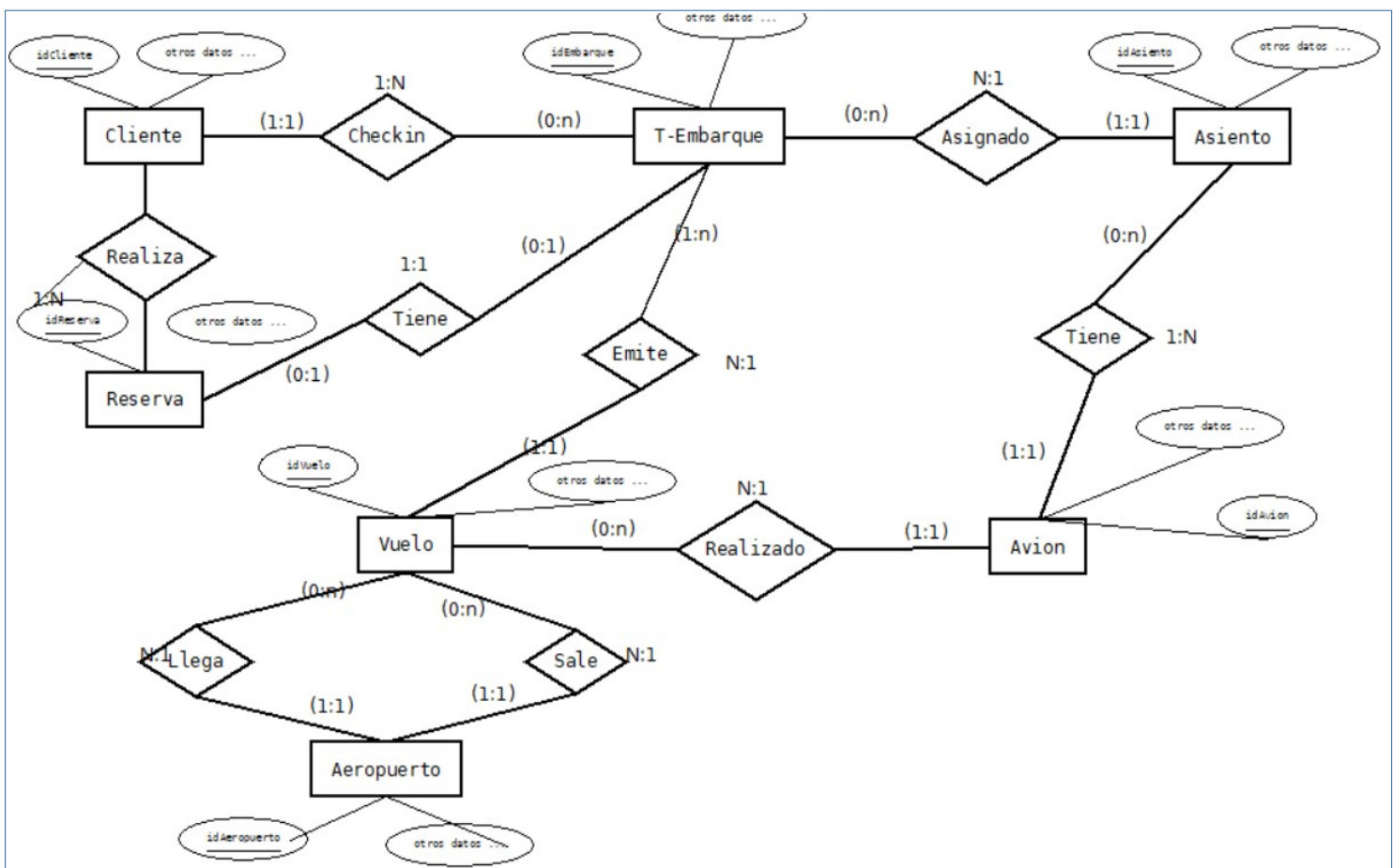


Ejercicio 3: Reserva de vuelos

Realizar un esquema entidad/relación (**notación Chen**) utilizando una herramienta similar a **Dia Diagram Editor** que permita gestionar reservas de vuelos, de modo que:

- Los clientes pueden reservar vuelos. Con la reserva se pueden reservar varias plazas, pero no poseeremos el número de asiento hasta obtener la tarjeta de embarque. En ese instante se asignará el asiento que tiene como identificación la fila, columna y la planta en la que está situado.
- Se pueden obtener tarjetas de embarque sin tener reserva.
- Las tarjetas de embarque se refieren a un único cliente. De modo que aunque reserváramos nueve plazas, cada cliente podrá sacar su tarjeta de embarque indicando el número de reserva, la fecha de la misma y sus datos personales (dni, nombre, apellidos, dirección y teléfono). Además la persona que reserva debe indicar una tarjeta de crédito que quedará asociada a esa persona.
- El vuelo que se reserva tiene un código único, una fecha y una hora de salida y de llegada y un aeropuerto de salida y otro de llegada.
- Los aeropuertos poseen un código único, además del nombre y la localidad y el país en el que se encuentran.
- Se guarda información sobre los aviones, código y número de plazas. Los vuelos sólo les puede realizar un avión determinado, pero el mismo avión puede realizar (como es lógico) otros vuelos.

MODELO CONCEPTUAL



PASO DEL MODELO CONCEPTUAL AL ESQUEMA RELACIONAL (NO GRÁFICO)

Cliente (idCliente, otros_datos, ...)

Reserva(idReserva, otros_datos, ..., idcliente, idTarjeta)

T_Embarque(idTarjeta, otros_datos, ..., idcliente, idAsiento)

Asiento(idAsiento, otros_datos, ..., idAvion)

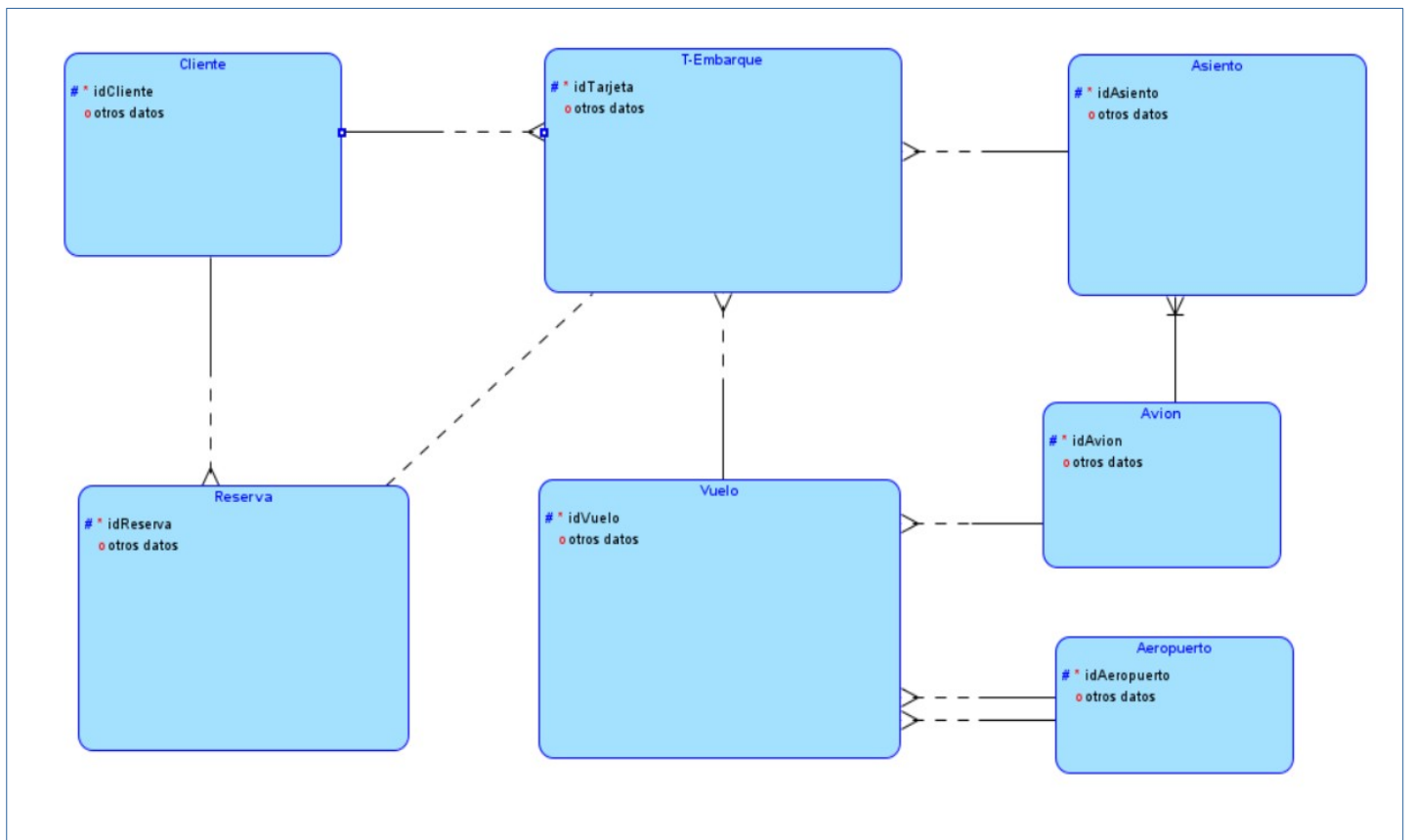
Avion(idAvion, otros_datos, ...)

Vuelo(idVuelo, otros_datos, ..., idAeroSalida, idAeroLlegada, idAvion)

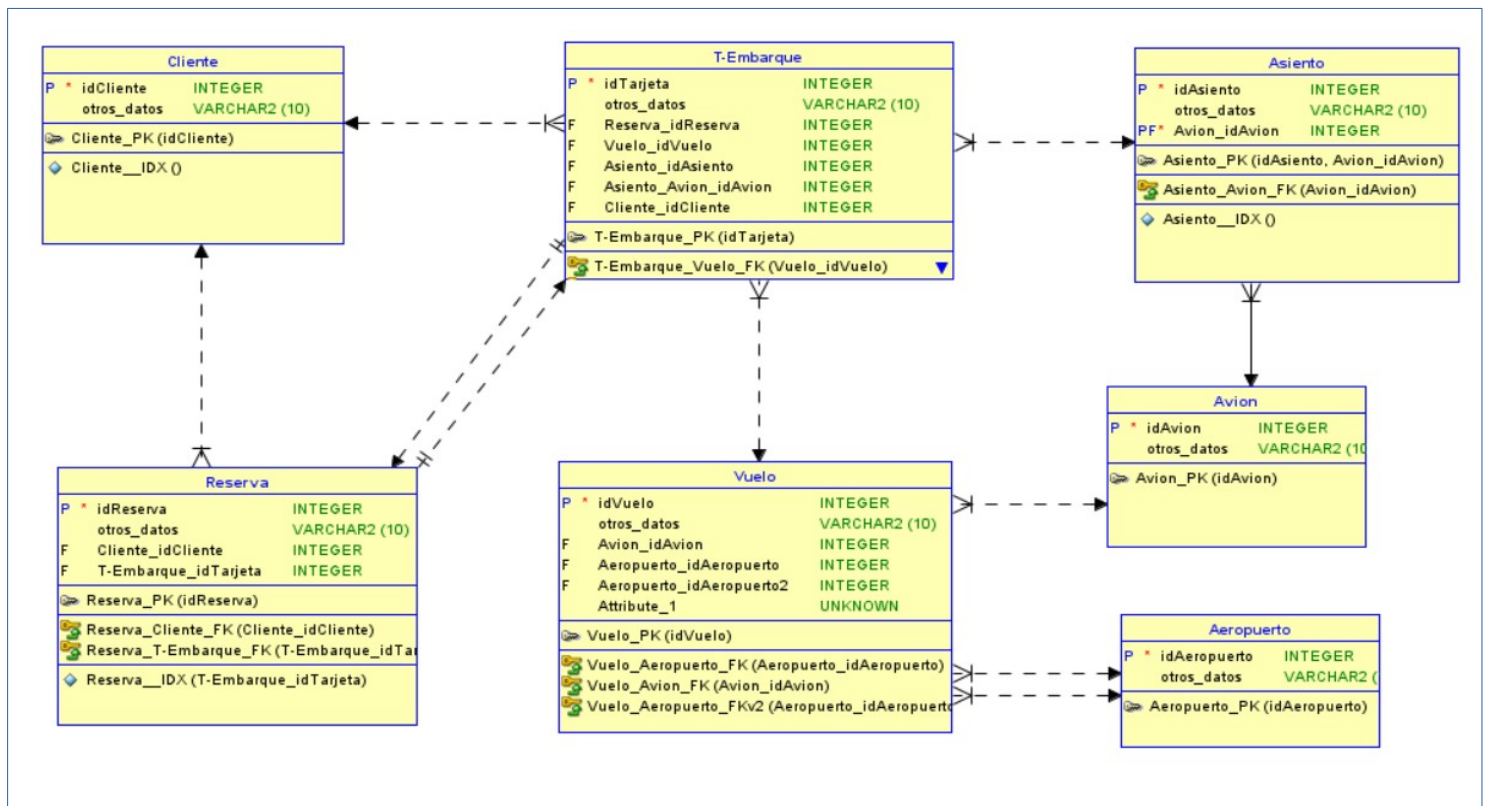
Aeropuerto (idAeropuerto, otros_datos, ...)

Ejercicio 4 : Realizar el ejercicio anterior utilizando la notación “pata de gallo” especificando las claves primarias y ajenas (foráneas) de cada relación (entidad).

Modelo lógico en Data Modeler



Modelo relacional en Data Modeler



Código SQL para crear la BD

```
CREATE TABLE aeropuerto (
    idaeropuerto INTEGER NOT NULL,
    otros_datos VARCHAR2(10)
);

ALTER TABLE aeropuerto ADD CONSTRAINT aeropuerto_pk PRIMARY KEY ( idaeropuerto );

CREATE TABLE asiento (
    idasiento          INTEGER NOT NULL,
    otros_datos        VARCHAR2(10),
    "T-Embarque_idTarjeta" INTEGER NOT NULL,
    avion_idavion      INTEGER NOT NULL
);

CREATE UNIQUE INDEX asiento__idx ON
    asiento (
        "T-Embarque_idTarjeta"
    ASC );

ALTER TABLE asiento ADD CONSTRAINT asiento_pk PRIMARY KEY ( idasiento,
    avion_idavion );

CREATE TABLE avion (
    idavion          INTEGER NOT NULL,
    otros_datos VARCHAR2(10)
```

```

);

ALTER TABLE avion ADD CONSTRAINT avion_pk PRIMARY KEY ( idavion );

CREATE TABLE cliente (
    idcliente            INTEGER NOT NULL,
    otros_datos          VARCHAR2(10),
    "T-Embarque_idTarjeta" INTEGER NOT NULL
);

CREATE UNIQUE INDEX cliente__idx ON
    cliente (
        "T-Embarque_idTarjeta"
    ASC );

ALTER TABLE cliente ADD CONSTRAINT cliente_pk PRIMARY KEY ( idcliente );

CREATE TABLE reserva (
    idreserva            INTEGER NOT NULL,
    otros_datos          VARCHAR2(10),
    cliente_idcliente    INTEGER,
    "T-Embarque_idTarjeta" INTEGER
);

CREATE UNIQUE INDEX reserva__idx ON
    reserva (
        "T-Embarque_idTarjeta"
    ASC );

ALTER TABLE reserva ADD CONSTRAINT reserva_pk PRIMARY KEY ( idreserva );

CREATE TABLE "T-Embarque" (
    idtarjeta            INTEGER NOT NULL,
    otros_datos          VARCHAR2(10),
    reserva_idreserva    INTEGER,
    vuelo_idvuelo        INTEGER
);

CREATE UNIQUE INDEX "T-Embarque__IDX" ON
    "T-Embarque" (
        reserva_idreserva
    ASC );

ALTER TABLE "T-Embarque" ADD CONSTRAINT "T-Embarque_PK" PRIMARY KEY ( idtarjeta );

CREATE TABLE vuelo (
    idvuelo              INTEGER NOT NULL,
    otros_datos          VARCHAR2(10),
    avion_idavion        INTEGER,
    aeropuerto_idaeropuerto INTEGER,
    aeropuerto_idaeropuerto1 INTEGER
);

ALTER TABLE vuelo ADD CONSTRAINT vuelo_pk PRIMARY KEY ( idvuelo );

ALTER TABLE asiento
    ADD CONSTRAINT asiento_avion_fk FOREIGN KEY ( avion_idavion )
        REFERENCES avion ( idavion );

ALTER TABLE asiento
    ADD CONSTRAINT "Asiento_T-Embarque_FK" FOREIGN KEY ( "T-Embarque_idTarjeta" )
        REFERENCES "T-Embarque" ( idtarjeta );

ALTER TABLE cliente
    ADD CONSTRAINT "Cliente_T-Embarque_FK" FOREIGN KEY ( "T-Embarque_idTarjeta" )

```



```

REFERENCES "T-Embarque" ( idtarjeta );

ALTER TABLE reserva
  ADD CONSTRAINT reserva_cliente_fk FOREIGN KEY ( cliente_idcliente )
    REFERENCES cliente ( idcliente );

ALTER TABLE "T-Embarque"
  ADD CONSTRAINT "T-Embarque_Vuelo_FK" FOREIGN KEY ( vuelo_idvuelo )
    REFERENCES vuelo ( idvuelo );

ALTER TABLE vuelo
  ADD CONSTRAINT vuelo_aeropuerto_fk FOREIGN KEY ( aeropuerto_idaeropuerto )
    REFERENCES aeropuerto ( idaeropuerto );

ALTER TABLE vuelo
  ADD CONSTRAINT vuelo_aeropuerto_fkv1 FOREIGN KEY ( aeropuerto_idaeropuerto1 )
    REFERENCES aeropuerto ( idaeropuerto );

ALTER TABLE vuelo
  ADD CONSTRAINT vuelo_avion_fk FOREIGN KEY ( avion_idavion )
    REFERENCES avion ( idavion );

```

Base de datos creada

