

FP055-Introduccion a las bases de datos

AA2. El diagrama Entidad - Relación

Contenido

Caso Practico.	4
1.1 SECCIÓN A. PARTE TEÓRICA	4
1.1.1 ¿Qué es un Modelo Lógico y qué se especifica en él?	4
1.1.2 Indicar la diferencia entre Entidad e Instancia y da dos ejemplos que ilustren estos conceptos.....	4
1.1.3 ¿Qué es un atributo y cuál es su objetivo?	5
1.1.4 ¿Qué es un Ideor Único (UID) y para qué se usa?	5
1.1.5 Dado un registro de clientes, proporcionar un ejemplo de atributo obligatorio y opcional y de atributo volátil y no volátil. Explicar por qué es importante seleccionar un atributo no volátil sobre uno volátil.	6
1.1.6 Mencionar los cuatro objetivos del Modelo de Relación de Entidades	7
1.1.7 ¿Qué es una relación entre entidades?.....	7
1.1.8 Indicar las dos palabras clave que te permitirán deducir la opcionalidad de una relación y da un ejemplo.	7
1.1.9 «La cardinalidad de una relación es el número de ocurrencias de una entidad asociadas a una ocurrencia de la otra entidad» (palabra clave: cuántos) ¿Cuántos tipos de cardinalidades hay? Define cada una de ellas y da un ejemplo.	8
1.1.10 ¿Es posible relacionar a una entidad con ella misma? Da un ejemplo	9
1.1.11 ¿Cuáles son los componentes de un ERD?	9
1.1.12 ¿Para qué se utilizan los Diagramas de Matriz?	10
1.1.13 Define supertipo y subtipo y da un ejemplo de cada uno.....	10
1.1.14 ¿A qué se llama Relación Redundante?	11
1.1.15 ¿A qué se denomina Entidad de Intersección?	11
1.1.16 ¿A qué se denomina UID compuesto? ¿Y UID artificial?	12
1.1.17 ¿Qué es un UID candidato?	12
1.1.18 ¿Qué es una relación de jerarquía? ¿Cómo se representa?.....	13
1.2 SECCIÓN B. PARTE PRÁCTICA.....	13
1.2.1 Punto 1.	13
1.2.2 Punto 2.	15
1.2.3 Punto 3.	15
1.2.4 Punto 4.	16
1.2.5 Punto 5.	16
1.2.6 Punto 6.	16
1.2.7 Punto 7.	16
1.2.8 Punto 8.	17
1.2.9 Punto 9.	17
1.2.10 Punto 10.	18

1.3	Realizar las actividades de la segunda parte de la actividad listadas a continuación:	18
1.3.1	Punto 1.	18
1.3.2	Punto 2.	18
1.3.3	Punto 3.	19
2	Bibliografía.	20

Caso Practico.

1.1 SECCIÓN A. PARTE TEÓRICA

1.1.1 ¿Qué es un Modelo Lógico y qué se especifica en él?

Se centra en la estructura lógica y la organización de los datos. Es la representación abstracta de una base de datos, como se organizarán y relacionarán los datos entre si dentro de la base de datos.

Dentro de este modelo se Especifica la estructura de la base de datos:

- Tablas.
- Relaciones.
- Atributos.
- Restricciones de integridad
- Índices.

El modelo lógico se especifica utilizando un lenguaje de modelado de datos, como el modelo Entidad-relación (ER) o el Modelo Relacional, ya que estos modelos proporcionan una representación visual de la estructura de la base de datos.

1.1.2 Indicar la diferencia entre Entidad e Instancia y da dos ejemplos que ilustren estos conceptos.

Para dar una correcta respuesta, hay que tener varios puntos en cuenta:

- Hay que tener en cuenta que una Entidad es una representación abstracta de un objeto o concepto en el mundo real que se modela en una base de datos. En cambio, las Instancias es una ocurrencia especifica de una Entidad en la base de datos.
- Las Entidades tienen atributos que describen sus propiedades o características, mientras que las instancias tienen los valores concretos para esos atributos.
- Las entidades pueden tener otras relaciones con otras entidades y si mismas dentro de la base datos. Mientras que las Instancias no pueden tener relación directa con otras instancias.
- Cada Entidad tiene una clave primaria (PRIMARY KEY) es el atributo que identifica sin posibilidad a equivocación, es único para esa entidad y no se puede repetir y no puede ser nulo en allá base de datos. Mientras que la Instancia tiene el valor específico para la clave primaria de su Entidad correspondiente.

UOC Universitat Oberta de Catalunya

Elementos del Modelo Relacional. ENTIDADES, INSTANCIAS, ATRIBUTOS

La **ENTIDAD** es la **unidad básica** del Modelo Relacional.
Representa cosas u objetos (ya sean reales o abstractos), que se diferencian claramente entre sí.
Está compuesta por **ATRIBUTOS**, los cuales describen sus características.

ENTIDAD: **INSTANCIA:** particularización u ocurrencia de una determinada entidad.

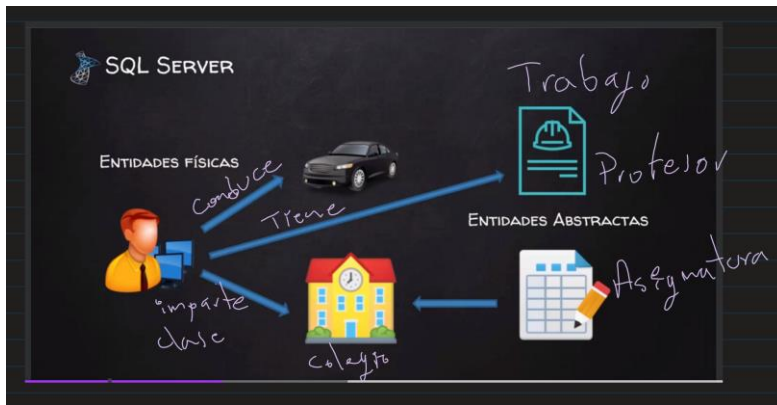
ACTOR

- # *id
- * nombre
- * apellido
- o web_oficial

ID	nombre	apellido	web_oficial
1	Angelina	Jolie	https://joliestweet.wixsite.com/angelinajolie
2	George	Clooney	http://www.gravitasdocufilms.com/george/

IDENTIFICADOR ÚNICO (UId)
El atributo (o combinación) que permite identificar **inequívocamente** la instancia de una entidad

II 4:22 / 11:33



1.1.3 ¿Qué es un atributo y cuál es su objetivo?

Es una propiedad o característica de una Entidad, y su objetivo es almacenar o representar un dato específico como por ejemplo el IDENTIFICADOR UNICO (UID) - PRIMARY KEY,



ENTIDAD	Ent	COCHE, TRABAJADOR, PEDIDO... (Objeto físico o Concepto Abstracto)
RELACIÓN	Rel	COMPRAR, PERTENECER, TENER... (Asociación de un Objeto con Otros Objetos)
ATRIBUTO	Atr	MATRÍCULA, DNI, COLOR... (Propiedades de la entidad)
PRIMARY KEY	<u>Atr</u>	<u>MATRÍCULA</u> , <u>DNI</u> , COLOR... (Es uno de los atributos que identifica sin posibilidad de equivocación a una entidad, quiere decir que debe ser único y debe existir, es decir no puede ser nulo en una base de datos). Estas condiciones se llaman restricciones.

1.1.4 ¿Qué es un Ideor Único (UID) y para qué se usa?

Es el Identificador unido.

PRIMARY KEY	<u>Atr</u>	<u>MATRÍCULA</u> , <u>DNI</u> , COLOR... (Es uno de los atributos que identifica sin posibilidad de equivocación a una entidad, quiere decir que debe ser único y debe existir, es decir no puede ser nulo en una base de datos). Estas condiciones se llaman restricciones.
-------------	------------	--

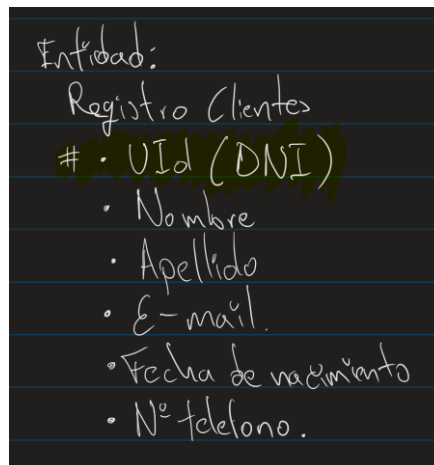
1.1.5 Dado un registro de clientes, proporcionar un ejemplo de atributo obligatorio y opcional y de atributo volátil y no volátil. Explicar por qué es importante seleccionar un atributo no volátil sobre uno volátil.

Dado el ejemplo de registro de cliente:

- Atributo UId: es identificador que permite identificar inequívocamente.
- Atributo obligatorio: atributo nombre y apellido, por ejemplo.
- Atributo Opcional: atributo E-mail, no todos quieren recibir información en el E-mail.
- Atributo no Volátil: Fecha de nacimiento. Es un atributo que no cambiara en el tiempo.
- Atributo volátil: Número de teléfono. Es un atributo que puede cambiar en el tiempo.

Hay que tener en cuenta que el Atributo UId es un Atributo no volátil por las siguientes razones:

- Integridad: hay que tener en cuenta que, si un UId se cambia en el tiempo, lo primero deja de ser UId y se pueden romper los enlaces o referencias a registros relacionados.
- Consistencia: Que garantice que el dato asociado al registro permanece constante a lo largo del tiempo.
- Desempeño y Fiabilidad: estos dos creo yo que van asociados a costes de desarrollo, ya que hacer cambios en un atributo no volátil.



#DNI	Nombre	Apellido	E-mail	Fecha de Nacimiento	Nº telefónico

❑ INTRODUCCIÓN AL MODELO RELACIONAL

- ❑ Los atributos pueden ser obligatorios u opcionales.
- ❑ Los obligatorios se representan con un "*"
- ❑ Los opcionales se representan con un "o"



1.1.6 Mencionar los cuatro objetivos del Modelo de Relación de Entidades

Los objetivos principales son los siguientes:

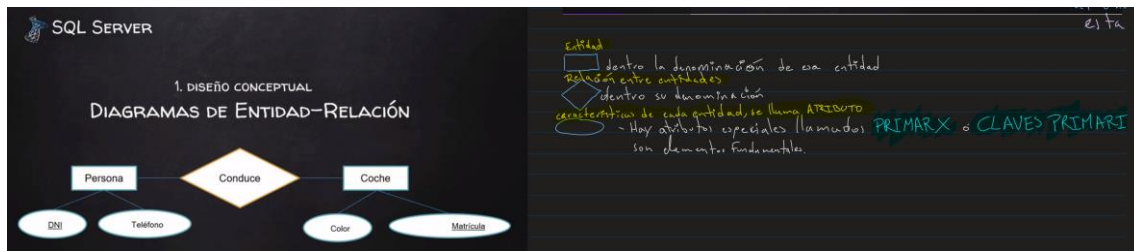
- Representación conceptual de la estructura de datos.
- Identificación de entidades y relaciones.
- Especificar restricciones de integridad.
- Facilitar la comunicación.

1.1.7 ¿Qué es una relación entre entidades?

Es la relación de una Entidad con otras entidades

• Un modelo de relación de entidades:

- Es una lista de todas las entidades y atributos, así como todas las relaciones entre las entidades que son importantes
- Proporciona información de fondo como, por ejemplo, descripciones de entidades, tipos de dato y restricciones
- Nota: El modelo no necesita un diagrama, pero el diagrama suele ser una herramienta muy útil



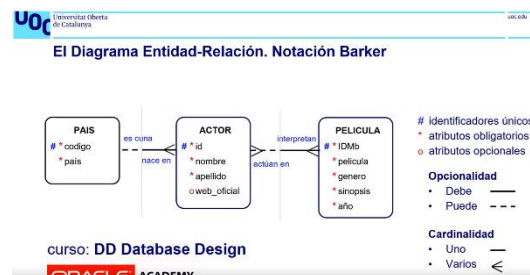
1.1.8 Indicar las dos palabras clave que te permitirán deducir la opcionalidad de una relación y da un ejemplo.

Opcionalidad

- Debe
- Puede

Cardinalidad

- Uno
- Varios



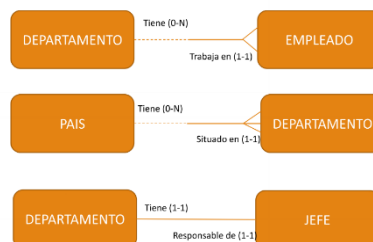
APRENDE SQL DESDE CERO

INTRODUCCIÓN AL MODELO RELACIONAL

- Las relaciones identifican la forma en la que las entidades se conectan unas con otras
- Se usan los siguientes elementos
 - Línea sólida ————— obligatoria
 - Línea punteada opcional
 - Tridente ≧ relación muchos

INTRODUCCIÓN AL MODELO RELACIONAL

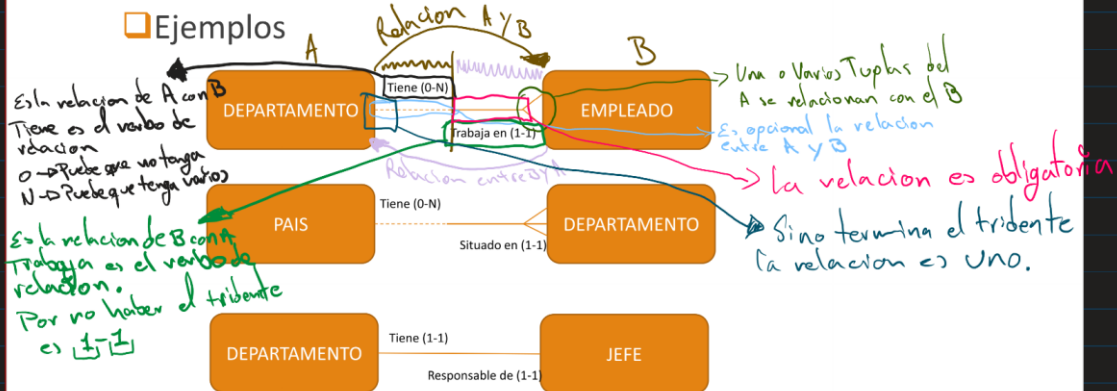
Ejemplos



APRENDE SQL DESDE CERO

INTRODUCCIÓN AL MODELO RELACIONAL

Ejemplos



1.1.9 «La cardinalidad de una relación es el número de ocurrencias de una entidad asociadas a una ocurrencia de la otra entidad» (palabra clave: cuántos) ¿Cuántos tipos de cardinalidades hay? Define cada una de ellas y da un ejemplo.

Cardinalidad.

Uno a Uno (1:1):

- Una instancia de la entidad A solo puede estar asociada con una instancia de entidad B, y viceversa.

Empleado - Asistente

Uno a Muchos (1:N) ó (1:M):

- Una instancia de la entidad A puede estar asociada con varias instancias de la entidad B, pero la B solo puede estar asociada con una instancia A.

Profesor - Estudiantes

Muchos a uno (N:1) ó (M:1):

Estudiantes - Escuela.

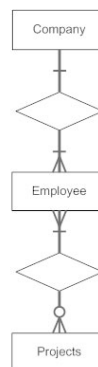
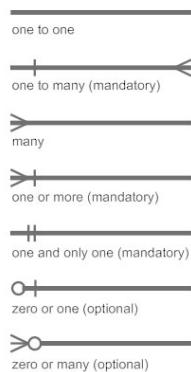
- Es lo inverso de 1:N

Muchos a Muchos (N:M)

- Una instancia de la entidad A puede estar asociada con varias instancias de la entidad B, y una instancia de B puede estar asociada con varias instancias de A.

Libro - Autores
Libros Autor.
Puede estar escrito por varios
Puede escribir varios.

Information Engineering Style



1.1.10 ¿Es posible relacionar a una entidad con ella misma? Da un ejemplo

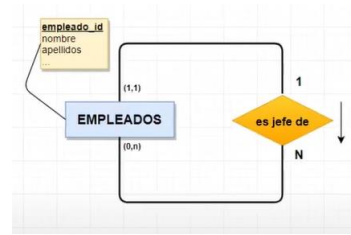
Si se puede, y se le llama relación recursiva o auto-relación.

Interrelación REFLEXIVA (o RECURSIVA):

Es una interrelación de una Entidad con ella misma.

Nota: una interrelación reflexiva se puede "interpretar" como una interrelación binaria de una entidad con una copia de sí misma. Y ya que "no se pueden repetir" las entidades, si se funde, queda la reflexiva

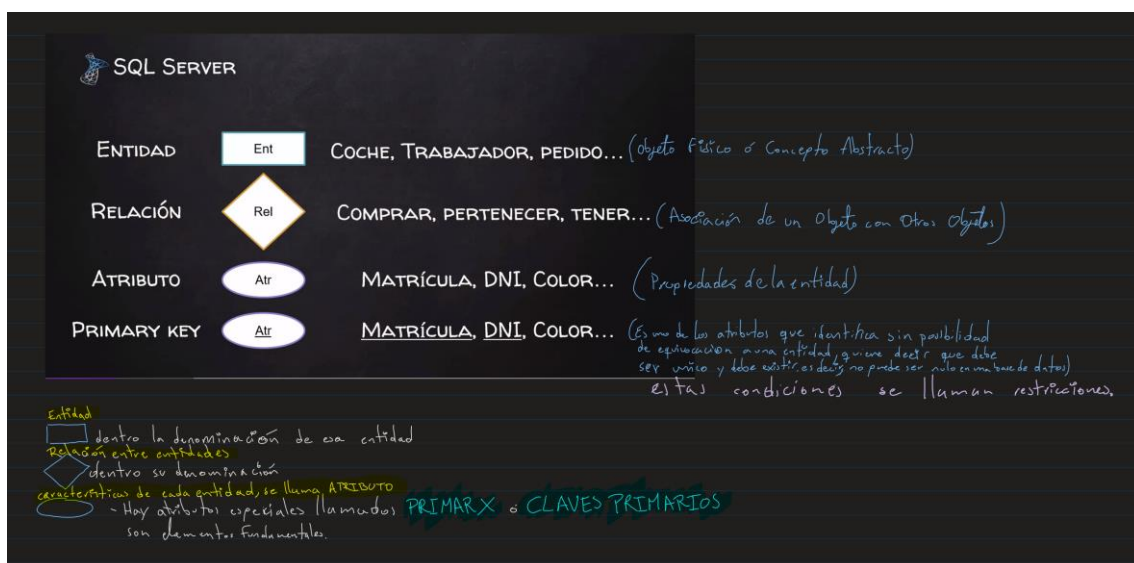
Las interrelaciones binarias recursivas pueden tener conectividad 1:1, 1:N o N:M, como todas las binarias.



1.1.11 ¿Cuáles son

los componentes de un ERD?

- Entidades:
 - Son los objetos o conceptos que tienen datos almacenados sobre ellos. Las entidades se pueden clasificar en Fuertes o débiles:
 - Entidades Fuertes: Tienen una existencia independiente y se representa por un rectángulo con el nombre de la entidad.
 - Entidad Débil: No pueden existir sin una entidad fuerte y esta conectada a una entidad fuerte mediante una relación y se representa con un rectángulo de doble línea.
- Atributos:
 - Son las propiedades o características de cada Entidad o relación que queremos almacenar en una base de datos. Estos atributos se representan con óvalos o círculos y se conectan con una entidad o relación con una línea y hay dos tipos de atributos:
 - Atributo simple: No se puede dividir en subpartes (ejemplo: número de teléfono de una persona).
 - Atributo compuesto: Puede ser dividido en subpartes más pequeñas (ejemplo: nombre completo de una persona)



- Cardinalidad: Indica las cantidades de instancias de una Entidad están relacionadas con las instancias de otra Entidad.

- Clave foránea (UId): es el atributo que tiene que tener una Entidad, que es único y sirve para la identificación sin posibilidad a equivocaciones.
- Relación: Es la representación de las asociaciones o conexiones entre las entidades.
- Restricciones: Son reglas o condiciones que se aplican a las relaciones entre entidades.

1.1.12 ¿Para qué se utilizan los Diagramas de Matriz?

Son herramientas de gestión y análisis que se utilizan para visualizar y analizar relaciones entre diferentes entidades o variables.

Algunas aplicaciones comunes de los diagramas de matriz incluyen:

- Análisis de riesgos: Para identificar y evaluar los riesgos asociados con diferentes variables o factores.
- Gestión de proyectos: Para asignar tareas y responsabilidades a los miembros del equipo y visualizar las dependencias entre ellas.
- Análisis de decisiones: Para evaluar y comparar diferentes opciones o alternativas en función de múltiples criterios.
- Planificación estratégica: Para identificar las fortalezas, debilidades, oportunidades y amenazas (FODA) de una organización o proyecto.
- Gestión de calidad: Para identificar y analizar las relaciones entre diferentes factores que pueden afectar la calidad de un producto o servicio.

1.1.13 Define supertipo y subtipo y da un ejemplo de cada uno.

Supertipo: Es una Entidad genérica que tiene un conjunto de atributos comunes que son compartidos por sus subtipos. El supertipo representa un concepto general o una categoría amplia.

Entidad Vehículo, Esta entidad tiene atributos comunes que podrían aplicarse a todos los diferentes tipos de vehículos:

- ID_Vehiculo.
- Fabricante.
- Modelo.
- Año.

Subtipo: Es una entidad especializada que hereda atributos del supertipo y puede tener atributos adicionales que son únicos para esa especialización. Los subtipos representan una versión más específica o una instancia del concepto general indicado por supertipo.

Supertipo: Vehículo → Subtipo:

- Automóvil:
 - Atributo adicional cantidad de puertas.
- Motocicleta:
 - Cilindrada.
- Camión:
 - Capacidad de carga.

1.1.14 ¿A qué se llama Relación Redundante?

Se refiere a una situación donde la información entre dos o más tablas se repite de manera innecesaria o donde las relaciones entre las tablas no están normalizadas de forma adecuada.

Esto puede llevar a varios problemas:

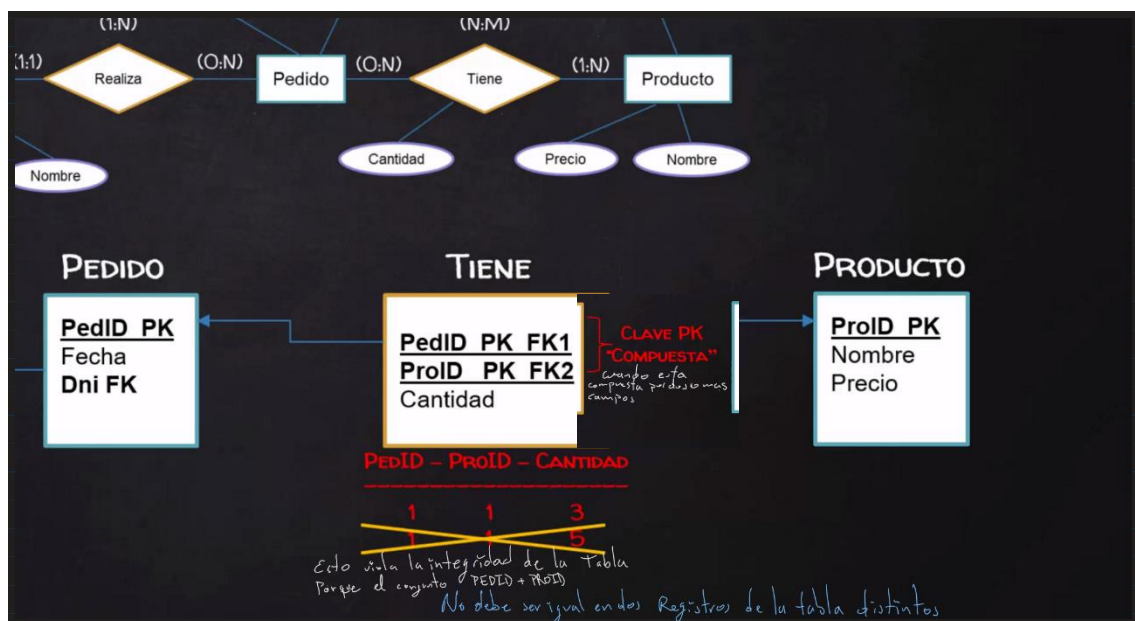
- La inconsistencia de datos.
- Dificultades en el mantenimiento de la base de datos.
- Uso ineficiente del espacio de almacenamiento.

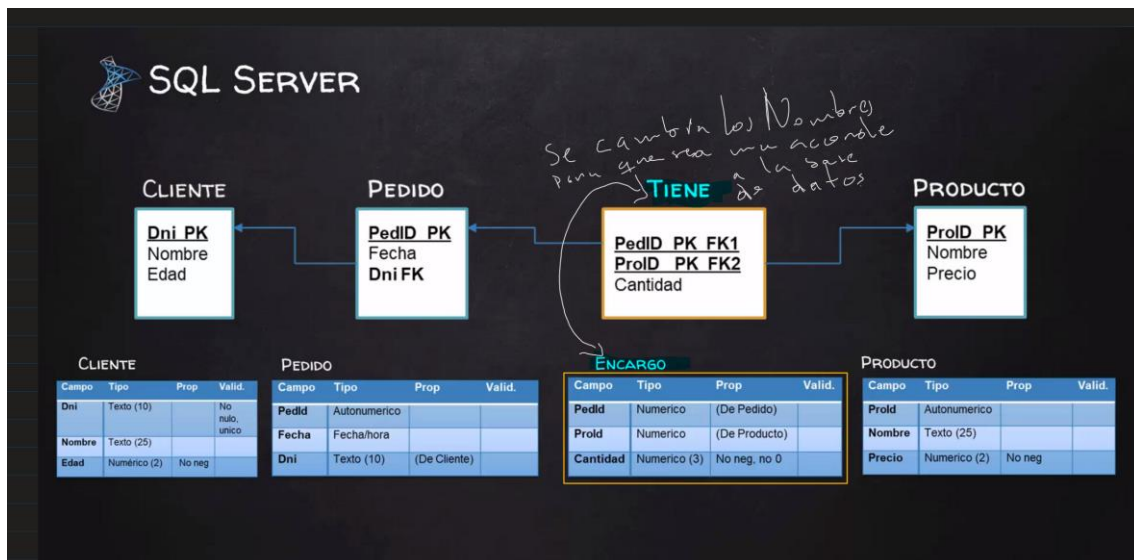
Esto lo puede ocasionar, por ejemplo:

- Datos duplicados en varias tablas que se relacionan.
- Relaciones innecesarias.

1.1.15 ¿A qué se denomina Entidad de Intersección?

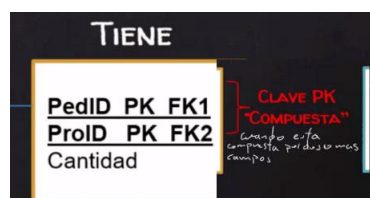
Es también conocida como entidad asociativa o tabla de unión, es utilizado para representar (N:M) entre dos entidades.





1.1.16 ¿A qué se denomina UID compuesto? ¿Y UID artificial?

UID compuesto: se refiere a una clave primaria que esta formada por la combinación de dos o mas atributos. Estos UID compuestos son comunes en intersecciones N:M



UID artificial: también conocido como clave sustituta, es un tipo de clave primaria que no tiene significado comercial o natural fuera de su uso como identificador en la base de datos. Este tipo de UID se genera automáticamente por la base de datos, a menudo como un número entero que se autoincrementa o un identificador único global (GUID). El propósito de un UID artificial es proporcionar un método simple y eficiente para identificar de manera única cada fila sin recurrir a datos que podrían cambiar o ser duplicados.

El uso de UIDs artificiales es común en situaciones donde no hay una clave natural obvia o donde la clave natural es voluminosa (compuesta por muchas columnas) o sensible a cambios, lo que podría afectar la integridad referencial si se utiliza como clave primaria.

1.1.17 ¿Qué es un UID candidato?

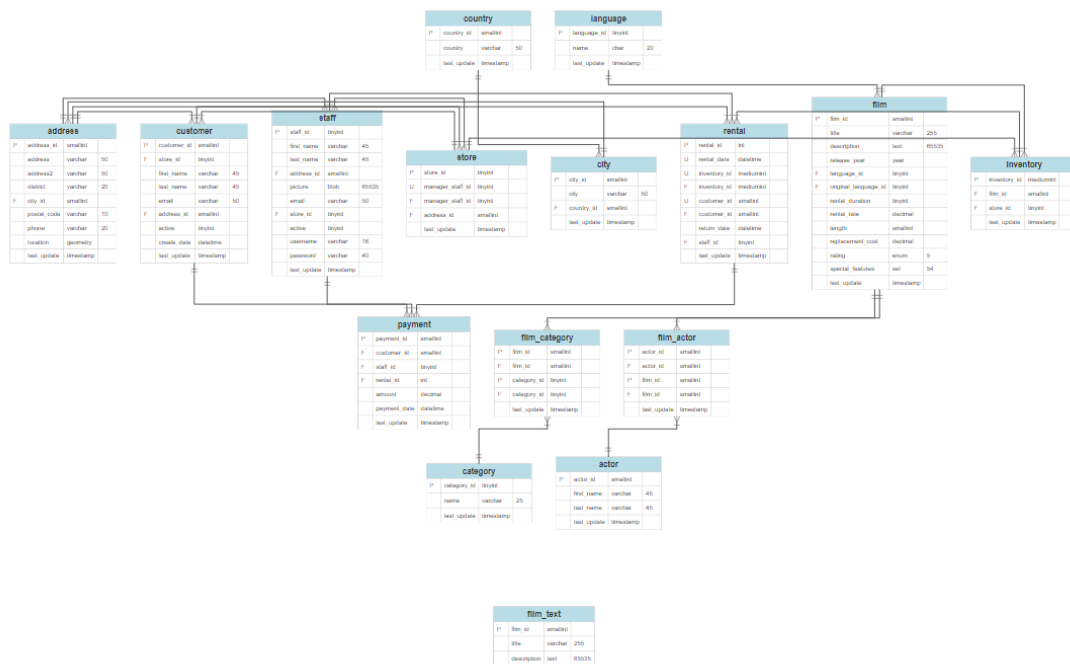
Es el conjunto de atributos que pueden servir de manera única para identificar un registro dentro de una tabla, es un atributo que sin ser UID es único e inequívoco.

Ejemplo:

- UID: DNI.
- Atributo 1: Seguridad social.
- Atributo 2: Carnet de conducir.
- Atributo 3: Cuenta Bancaria.

1.1.18 ¿Qué es una relación de jerarquía? ¿Cómo se representa?

Se refiere a la estructura de datos donde los elementos están organizados en niveles o capas, donde cada elemento o capara tiene un único “Padre” y este puede tener varios hijos.



Leer y comprender la parte práctica de la segunda parte de la actividad indicado a continuación:

1.2 SECCIÓN B. PARTE PRÁCTICA

1.2.1 Punto 1.

Se desea crear una base de datos para gestionar a los socios de un espacio de coWorking y los diferentes espacios y actividades que se ofrecen. A continuación, se explican los requerimientos de la misma:

El espacio coWorking cuenta con 7 PLANES, cinco de ellos dedicados a autonomos (A) y los dos últimos de ellos dirigidos a empresas(E). Se desea guardar para cada plan, su nombre, la inscripción por persona, el coste mensual y su descripción.

Se adjunta la tabla de planes, para un mayor entendimiento del caso:

ID	Planes	Tipo	Matrícula persona	Cuota mensual	Descripción
1	BÁSICO MAÑANAS	A	45,00€	40,00€	Acceso de 8:00 a 14:00
2	BÁSICO TARDES	A	45,00€	40,00€	Acceso de 15:00 a 21:00
3	DIURNO	A	45,00€	50,00€	Acceso de 9:00 a 18:00
4	24h	A	45,00€	70,00€	Acceso 24h

ID	Planes	Tipo	Matrícula persona	Cuota mensual	Descripción
5	Joven Emprendedor <24 años	A	0€	35,00€	Cualquier hora máximo 6h diarias
6	StartUp	E	0€	60,00€	Acceso de 9:00 a 18h
7	StartUp 24h	E	0€	80,00€	Acceso 24h

```

1  -- Comando para borrar tabla ya creada //////////DROP Table PLANES//////////
2  --Esta es la secuencia para crear la table en comando SQL//
3  -- Creación de la tabla Planes
4
5  CREATE TABLE Planes (
6      ID INT PRIMARY KEY,
7      Nombre VARCHAR2(255),
8      Tipo CHAR(1) CHECK (Tipo IN ('A', 'E')),
9
10     MatriculaPersona DECIMAL(7, 2),
11     CuotaMensual DECIMAL(7, 2),
12     Descripcion VARCHAR2(1000)
13 );

```

```

1  --Para insertar datos en una tabla concreta, se hace de la siguiente manera//
2  INSERT ALL
3  INTO Planes (ID, Nombre, Tipo, MatriculaPersona, CuotaMensual, Descripcion) VALUES (1, 'BÁSICO MAÑANAS', 'A', 45.00, 40.00, 'Acceso de 8:00 a 14:00')
4  INTO Planes (ID, Nombre, Tipo, MatriculaPersona, CuotaMensual, Descripcion) VALUES (2, 'BÁSICO TARDES', 'A', 45.00, 40.00, 'Acceso de 15:00 a 21:00')
5  INTO Planes (ID, Nombre, Tipo, MatriculaPersona, CuotaMensual, Descripcion) VALUES (3, 'DIURNO', 'A', 45.00, 50.00, 'Acceso de 9:00 a 18:00')
6  INTO Planes (ID, Nombre, Tipo, MatriculaPersona, CuotaMensual, Descripcion) VALUES (4, '24h', 'A', 45.00, 70.00, 'Acceso 24h')
7  INTO Planes (ID, Nombre, Tipo, MatriculaPersona, CuotaMensual, Descripcion) VALUES (5, 'Joven Emprendedor <24 años', 'A', 0.00, 35.00, 'Cualquier hora máximo 6h diarias')
8  INTO Planes (ID, Nombre, Tipo, MatriculaPersona, CuotaMensual, Descripcion) VALUES (6, 'StartUp', 'E', 0.00, 60.00, 'Acceso de 9:00 a 18h')
9  INTO Planes (ID, Nombre, Tipo, MatriculaPersona, CuotaMensual, Descripcion) VALUES (7, 'StartUp 24h', 'E', 0.00, 80.00, 'Acceso 24h')
10 SELECT * FROM DUAL;
11

```

1.2.2 Punto 2.

Existen dos tipos de trabajadores o socios que tendrán acceso al espacio coWorking:

- **Trabajador Corporativo:** Afiliado por una organización a un Plan de Empresa. Además de los datos comunes a todos los socios, se le registra el NIF de la empresa a la que pertenece.
- **Trabajador Autonomo:** Es la persona responsable del pago de un determinado Plan. Además de los datos comunes se le registra el Banco y la Cuenta donde se ha de domiciliar el pago

```
1  -- Comando para borrar tabla ya creada //////DROP Table PLANES////////
2  -- Esta es la secuencia para crear la table en comando SQL*/
3  -- Creación de la tabla Planes
4  CREATE TABLE Planes (
5      ID INT PRIMARY KEY,
6      Nombre VARCHAR2(255),
7      Tipo CHAR(1) CHECK (Tipo IN ('A', 'E')),
8      MatriculaPersona DECIMAL(7, 2),
9      CuotaMensual DECIMAL(7, 2),
10     Descripcion VARCHAR2(1000)
11 );
12
13 -- Creación de la tabla Socios para almacenar datos comunes de todos los socios
14
15 CREATE TABLE Socios (
16     ID_Socio INT PRIMARY KEY,
17     Nombre VARCHAR2(255),
18     Direccion VARCHAR2(255),
19     Telefono VARCHAR2(15),
20     Email VARCHAR2(100),
21     ID_Plan INT,
22     FechaInscripcion DATE,
23     FechaCaducidad DATE,
24     CONSTRAINT fk_plan FOREIGN KEY (ID_Plan) REFERENCES Planes(ID)
25 );
26
27 -- Creación de la tabla TrabajadoresCorporativos para aquellos afiliados por una organización
28
29 CREATE TABLE TrabajadoresCorporativos (
30     ID_Socio INT PRIMARY KEY,
31     NIF_Empresa VARCHAR2(9),
32     CONSTRAINT fk_socio_corporativo FOREIGN KEY (ID_Socio) REFERENCES Socios(ID_Socio)
33 );
34
35 -- Creación de la tabla TrabajadoresAutonomos para los responsables del pago del plan
36
37 CREATE TABLE TrabajadoresAutonomos (
38     ID_Socio INT PRIMARY KEY,
39     Banco VARCHAR2(255),
40     NumeroCuenta VARCHAR2(14),
41     CONSTRAINT fk_socio_autonomo FOREIGN KEY (ID_Socio) REFERENCES Socios(ID_Socio)
42 );
```

1.2.3 Punto 3.

Los datos comunes a cada socio son: Id_Socio (generado por el espacio coWorking), Documento de Identidad, Nombre, Apellido1, Apellido2, Sexo, Fecha de Nacimiento, Plan (de los 7 de la tabla), Fecha Alta, Número de Tarjeta de Acceso, Teléfono, Email, Ciudad, Código Postal y Observaciones.

El espacio coWorking no desea perder ningún dato de los socios por lo cual, a los datos comunes se agrega un campo que indique si el socio está activo o no.

```
46  --Una vez creadas las tablas, solo he encontrado dos soluciones o borrar la tabla entera con la primera línea o modificar y agregar, como lo hago a continuacion*/
47  ALTER TABLE Socio ADD DocumentoIdentidad VARCHAR2(20);
48  ALTER TABLE Socio ADD Apellido1 VARCHAR2(255);
49  ALTER TABLE Socio ADD Apellido2 VARCHAR2(255);
50  ALTER TABLE Socio ADD Sexo CHAR(1) CHECK (Sexo IN ('M','F','O'));
51  ALTER TABLE Socio ADD FechaNacimiento DATE;
52  ALTER TABLE Socio ADD FechaAlta DATE;
53  ALTER TABLE Socio ADD NumeroTarjetaAcceso VARCHAR2(20);
54  ALTER TABLE Socio ADD Ciudad VARCHAR2(255);
55  ALTER TABLE Socio ADDCodigoPostal VARCHAR(10);
56  ALTER TABLE Socio ADD Observaciones VARCHAR2(1000);
57  ALTER TABLE Socio ADD Activo CHAR(1) check (Activo IN('S', 'N'));
58
59  -- Creación de la secuencia para el ID_Socio automáticamente por sistema.
60  CREATE SEQUENCE socio_seq START WITH 1;
61
62  -- CREATE OR REPLACE TRIGGER Socios_before_insert
63  BEFORE INSERT ON Socios
64  FOR EACH ROW
65  BEGIN
66      SELECT socio_seq.NEXTVAL INTO :new.ID_Socio FROM dual;
67  END;
```


1.2.4 Punto 4.

Se desea un registro de EMPRESAS que tengan convenios con el espacio coWorking para planes CORPORATIVOS, donde se guarde: NIF, Nombre de la Empresa, teléfono de contacto, persona de contacto, mail, Fecha inicio convenio, Fecha fin convenio.

```
70
71
72 CREATE TABLE Empresas (
73     NIF VARCHAR2(9) PRIMARY KEY,
74     NombreEmpresa VARCHAR2(255),
75     TelefonoContacto VARCHAR2(15),
76     PersonaContacto VARCHAR2(255),
77     EmailContacto VARCHAR2(100),
78     FechaInicioConvenio DATE,
79     FechaFinConvenio DATE
80 );
-- NIF de la empresa, que actúa como clave primaria única
-- Nombre de la empresa
-- Teléfono de contacto con la empresa
-- Nombre de la persona de contacto en la empresa
-- Correo electrónico de la persona de contacto
-- Fecha de inicio del convenio con el coWorking
-- Fecha en la que termina el convenio
```

1.2.5 Punto 5.

Dado que los trabajadores autónomos pueden ir cambiando de planes con el tiempo o darse de baja definitivamente, cada vez que esto ocurra, se desea llevar un registro HISTÓRICO de planes (Audit Trail) por Socio, donde se indique: idSocio, FechaCambio, PlanAnterior, PlanActual, Fecha Alta Plan Anterior y si se trata o no de una Baja Definitiva.

```
81
82 CREATE TABLE HistoricoPlanesSocios (
83     ID_Historico INT PRIMARY KEY,
84     ID_Socio INT,
85     FechaCambio DATE,
86     PlanAnterior INT,
87     PlanActual INT,
88     FechaAltaPlanAnterior DATE,
89     BajaDefinitiva CHAR(1) CHECK (BajaDefinitiva IN ('S', 'N')),
90     CONSTRAINT fk_historico_socio FOREIGN KEY (ID_Socio) REFERENCES Socios(ID_Socio),
91     CONSTRAINT fk_plan_anterior FOREIGN KEY (PlanAnterior) REFERENCES Planes(ID),
92     CONSTRAINT fk_plan_actual FOREIGN KEY (PlanActual) REFERENCES Planes(ID)
93 );
-- Un identificador único para cada entrada en el historial
-- La clave foránea que se refiere al ID del socio en la tabla Socios
-- La fecha en la que se realizó el cambio de plan o la baja
-- La clave foránea que se refiere al plan anterior del socio
-- La clave foránea que se refiere al nuevo plan del socio
-- La fecha de alta del plan anterior
-- Indica si se trata de una baja definitiva ('S' para sí, 'N' para no)
```

1.2.6 Punto 6.

El espacio coWorking ofrece diferentes cursos y seminarios (incluidos según el Plan contratado).

Para cada actividad (curso o seminario) se desea registrar: IdActividad, Actividad, Descripción, Nivel [Inicial / Básico / Intermedio / Avanzado], duración de la sesión en minutos y varios campos que se deben seleccionar en caso de que la actividad los incluya. Estos son: Certificado oficial, Certificado del curso, Presencial y Materiales propios, entendiendo que una actividad tener más de una de estas características. Además, dejar un espacio para Observaciones.

```
94 CREATE TABLE Actividades (
95     IdActividad INT PRIMARY KEY,
96     Actividad VARCHAR2(255),
97     Descripcion VARCHAR2(1000),
98     Nivel VARCHAR2(50) CHECK (Nivel IN ('Inicial', 'Básico', 'Intermedio', 'Avanzado')),
99     DuracionSesion INT,
100     CertificadoOficial CHAR(1) CHECK (CertificadoOficial IN ('S', 'N')),
101     CertificadoCurso CHAR(1) CHECK (CertificadoCurso IN ('S', 'N')),
102     Presencial CHAR(1) CHECK (Presencial IN ('S', 'N')),
103     MaterialesPropios CHAR(1) CHECK (MaterialesPropios IN ('S', 'N')),
104     Observaciones VARCHAR2(1000)
105 );
-- Duración de la sesión en minutos
eto. Se pueden usar comillas dobles
ingles/minúsculas.
```

1.2.7 Punto 7.

Para las actividades se llenará un HORARIO, que tendrá los siguientes campos: id_Actividad, Sala (instalación), Fecha, Hora, Profesor, Observaciones. Habrá que tener en cuenta que las actividades no se solapen en una misma instalación, ya que el horario será publicado en la web.

```
96
97 CREATE TABLE Horarios (
98     IdHorario INT PRIMARY KEY,
99     IdActividad INT NOT NULL,
100     Sala VARCHAR2(255) NOT NULL,
101     Fecha DATE NOT NULL,
102     Hora VARCHAR2(5) NOT NULL, -- Formato 'HH24:MI'
103     Profesor VARCHAR2(255),
104     Observaciones VARCHAR2(1000),
105     CONSTRAINT fk_horario_actividad FOREIGN KEY (IdActividad) REFERENCES Actividades (IdActividad)
106 );
107
108
```


1.2.10 Punto 10.

Los socios podrán solicitar un SEGUIMIENTO personalizado. Para aquellos socios que lo soliciten, se les llevará un registro donde se guardará: idSocio, Fecha de realización de la actividad y la nota obtenida.

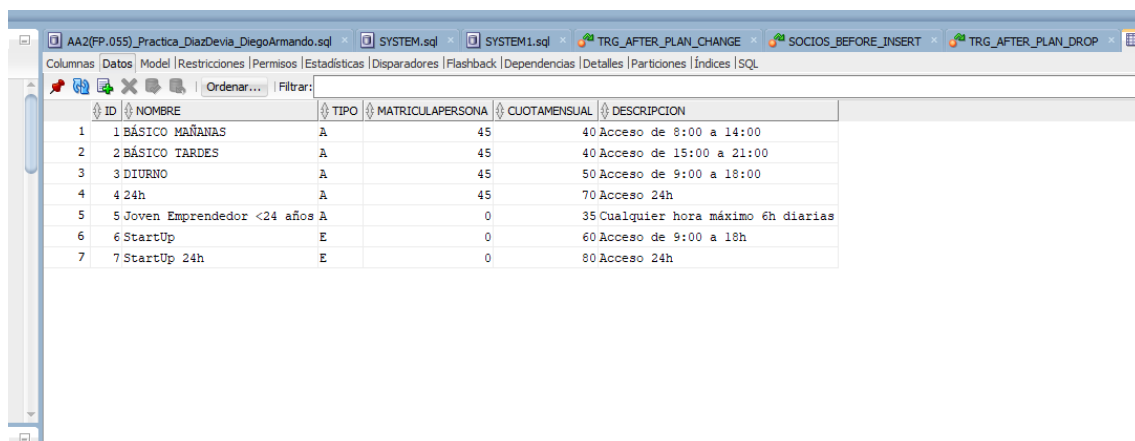
```
141 CREATE TABLE Seguimiento (  
142     idSeguimiento INT PRIMARY KEY,  
143     idSocio INT NOT NULL,  
144     FechaActividad DATE NOT NULL,  
145     NotaObtenida NUMBER(3,1),  
146     CONSTRAINT fk_seguimiento_socio FOREIGN KEY (idSocio) REFERENCES Socios (ID_Socio)  
147 );  
148  
149 --Se crea una secuencia para los ID de seguimiento  
150  
151 CREATE SEQUENCE seguimiento_seq START WITH 1;  
152
```

1.3 Realizar las actividades de la segunda parte de la actividad listadas a continuación:

1.3.1 Punto 1.

Identificar en una tabla como la del siguiente ejemplo las entidades, atributos e identificador principal de cada entidad del caso práctico.

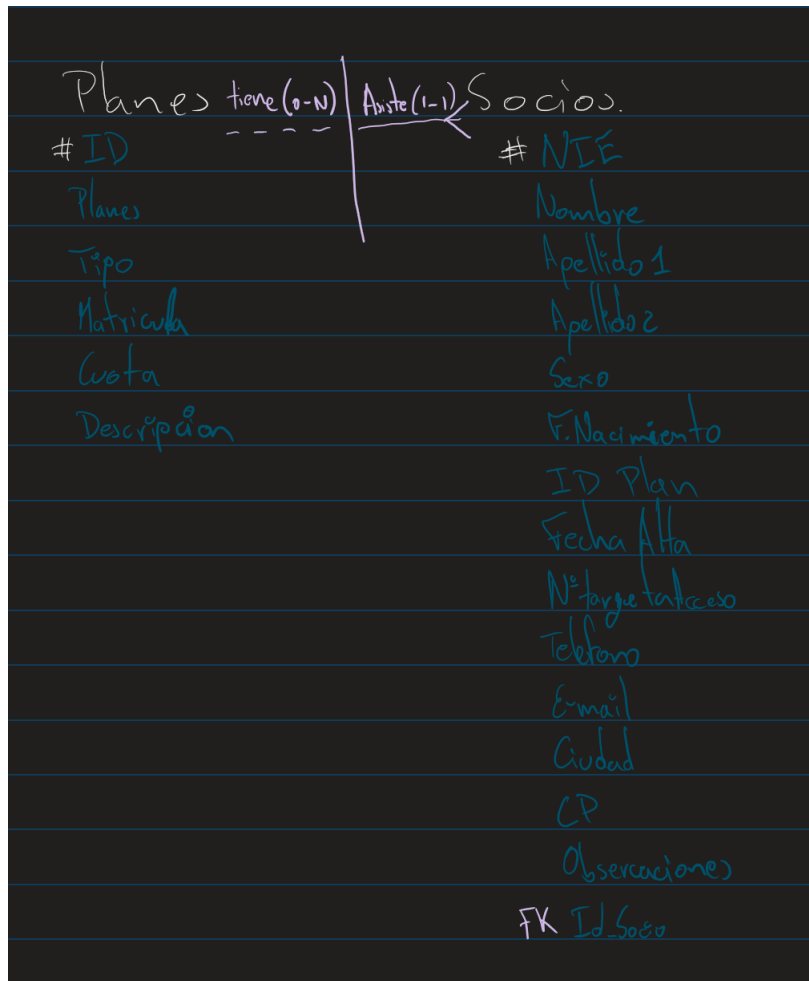
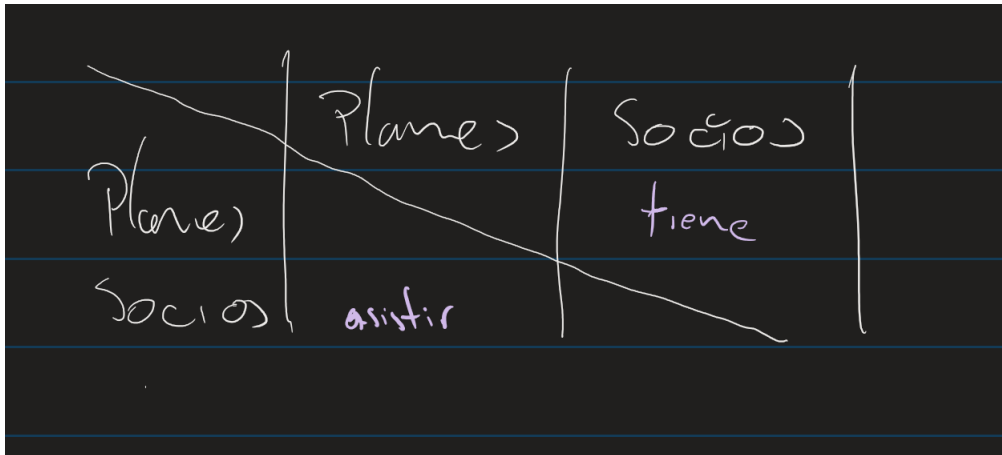
Entidad	Atributos	Identificador principal
ACTORES	ID, nombre, apellidos, país, web_oficial	ID



ID	NOMBRE	TIPO	MATRICULAPERSONA	CUOTAMENSUAL	DESCRIPCION
1	BÁSICO MAÑANAS	A	45	40	Acceso de 8:00 a 14:00
2	BÁSICO TARDES	A	45	40	Acceso de 15:00 a 21:00
3	DIURNO	A	45	50	Acceso de 9:00 a 18:00
4	24h	A	45	70	Acceso 24h
5	Joven Emprendedor <24 años	A	0	35	Cualquier hora máximo 6h diarias
6	StartUp	E	0	60	Acceso de 9:00 a 18h
7	StartUp 24h	E	0	80	Acceso 24h

1.3.2 Punto 2.

Utilizar un diagrama de matriz para identificar las relaciones existentes entre las diferentes entidades del caso práctico anterior. Encontrarás un ejemplo en el Curso DD Database Design. DD 3-4 Diagramas de Matriz del curso de Oracle Academy.



1.3.3 Punto 3.

Adjunto el PDF con el modelo exportado directamente de DATAMODELER.

2 Bibliografía.

1.1.1. Que es base de datos e información general.

[Lenguaje de consulta estructurado \(SQL\) - GeeksforGeeks](#)
[SQL Tutorial \(geeksforgeeks.org\)](#)

1.1.2. Conceptos de las base de datos.

[Introduction to Oracle Database](#)

1.1.3. Información sobre ERD.

[¿Qué es el Diagrama Entidad-Relación \(ERD\)? \(visual-paradigm.com\)](#)
[Entity Relationship Diagram \(ERD\) - What is an ER Diagram? \(smartdraw.com\)](#)

1.1.4. Símbolos y notificación del diagrama entidad-relación

[Símbolos y notación del diagrama entidad-relación | Lucidchart](#)

1.1.5. Mas información sobre diagramas de Matriz

[Diagrama matricial, ¿qué es? \(edrawsoft.com\)](#)

1.1.6. Curso SQL UDEMY

[https://www.udemy.com/share/101HMI3@u2Pvzp5MPadfKNtbjpmHNIj2Gbr2LD](#)
[MShAkim5P9kAjNmrXw-_oulwokLi5g04Sw/](#)