

UT2. Diseño conceptual. El modelo entidad-relación

¿Cuál es uno de los principales motivos del fracaso en el diseño de una aplicación?

La respuesta a esta pregunta sería la poca confianza en las metodologías de diseño de bases de datos. Esta desconfianza tiene sus consecuencias:

- Se subestima el tiempo o los recursos necesarios
- Las bases de datos son inadecuadas o insuficientes
- La documentación es limitada
- El mantenimiento es complicado

Por eso es importante seguir los pasos para realizar un buen diseño de la base de datos:

Especificación de requisitos

DISEÑO CONCEPTUAL

Esquema conceptual

DISEÑO LÓGICO

Esquema lógico

DISEÑO FÍSICO

Esquema físico

En la etapa del **diseño conceptual** se busca la manera de representar la estructura de la base de datos, **normalmente relacional**, buscando la representación de esquemas.

En 1976 y 1977 dos artículos de Peter P. Chen detallaron un modelo para realizar esquemas con la idea de proveer una visión unificada de los datos de un sistema de base de datos. **Este modelo es el modelo entidad/interrelación (entity/relationship en inglés) que actualmente se conoce más con el nombre de entidad/relación (Modelo E/R o ME/R, en inglés E/RM).**

Posteriormente **otros autores han añadido mejoras a este modelo, lo que ha producido toda una familia de modelos basados en el modelo Entidad/Relación original.** La más aceptada actualmente es el modelo **entidad/relación extendido (ERE)** que complementa algunas carencias del modelo original. **No obstante las diversas variantes del modelo hacen que los esquemas que dibujan los profesionales no sigan un verdadero estándar y sean dispares, aunque hay ideas muy comunes a todos los “dialectos” del modelo entidad/relación.**

Hay que insistir en que este modelo no tiene nada (o muy poco) que ver con las bases de datos relacionales, los esquemas entidad/relación se pueden utilizar (en principio) con cualquier SGBD ya que son conceptuales. Puede que nos confunda el uso de la palabra relación, ya que está también presente en el modelo relacional de las bases de datos de Edgar F. Codd; pero el concepto de relación en este esquema no tiene nada que ver con la idea de relación expuesta por Codd en su modelo relacional

Estos diagramas ayudan a entender los datos y cómo se relacionan entre ellos. Los principales elementos que componen este modelo son las entidades, los atributos simples, las relaciones, la cardinalidad entre las relaciones, los atributos que son clave primaria y los atributos que son clave ajena o foránea.

2.1. Diseño conceptual de la base de datos

La intención del modelado de datos es estructurar y organizar los datos para que se puedan utilizar fácilmente por la base de datos. El modelo conceptual en el diseño de bases de datos pretende representar la información que queremos almacenar de una manera normalizada. Para ello se fijará en 4 puntos:

- **Detección de las entidades:** detectar cada elemento del que se quiere guardar información
- **Detección de atributos:** reconocer las propiedades o características de esas entidades
- **Detección de relaciones:** qué relaciones existen entre las entidades
- **Detección de restricciones:** conocer las posibles restricciones de cada una de las propiedades de cada tipo de entidad, tanto para los atributos como para las relaciones

El modelo de datos depende de la visión que tenga el diseñador de la base de datos de los requisitos.

2.1.1 Entidades

Las entidades serán los conceptos de interés en nuestra base de datos, pueden ser **objetos reales** (vehículos, personas, animales, ...) o **abstractos** (cursos, asignaturas, profesiones, ...)

Uno de los problemas en el diseño conceptual es tomar la decisión de si un determinado concepto u objeto es o no una entidad, ¿podría ser un atributo? Para ayudarnos en la decisión basta con tener en cuenta que para que un concepto/objeto sea modelado como entidad debe cumplir 3 reglas:

1. Debe tener existencia propia
2. Cada ejemplar de una entidad debe poder distinguirse de otro
3. Todos los ejemplares de una entidad deben tener las mismas propiedades

Ejemplos: profesor, estudiante, asignatura, coche, libro, tienda, usuario, móviles, ...

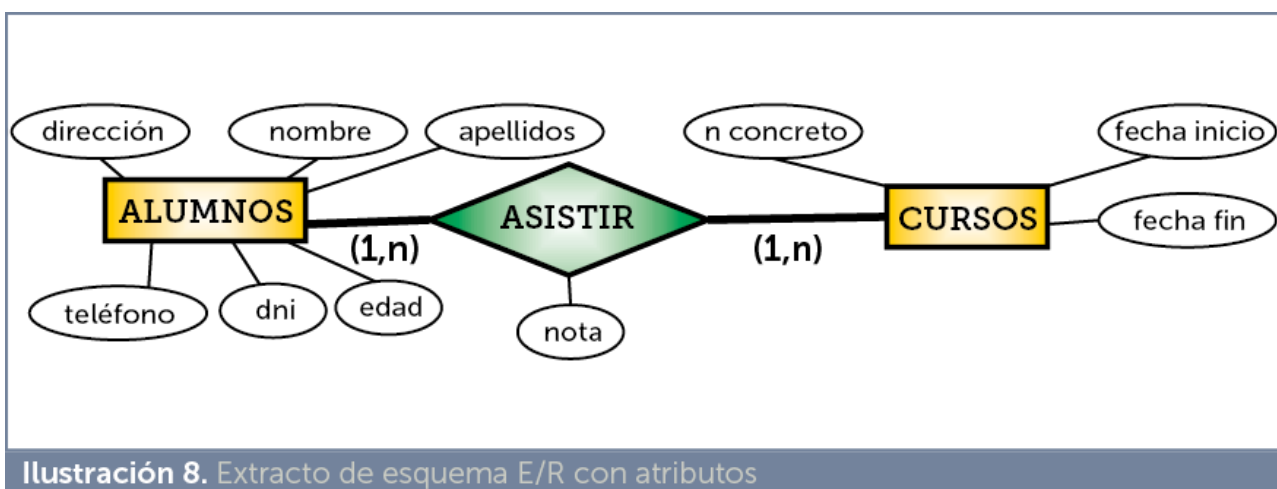
La representación en un modelo entidad-relación de una entidad es un rectángulo que en el interior lleva el nombre de la entidad:

PROFESOR		ESTUDIANTE		ASIGNATURA		COCHE
LIBRO		TIENDA		USUARIO		MÓVILES

2.1.2 Atributos

Los atributos definen las características de una entidad. Podríamos decir que los atributos describen a la entidad y pueden ser de distintos tipos. Por ejemplo, para la entidad **PROFESOR**, posibles atributos serían: *DNI, Nombre, primer apellido, segundo apellido, nº seguridad social, fecha nacimiento, domicilio, etc ...*

El modelo Entidad/Relación clásico los representa con **elipses**, dentro de las cuales se coloca el nombre del atributo. La elipse se une con una línea a las entidades. Ejemplo:



Debemos tener en cuenta que a veces un aparente atributo es en realidad una entidad, en general podemos afirmar que una propiedad es una entidad si ella misma tiene sus propias propiedades.

PROFESOR	ESTUDIANTE	ASIGNATURA	COCHE
DNI Nombre Apellido1 Apellido2 NumSeguridadSocial	DNI Nombre Apellido1 Apellido2 FechaNacimiento	Nombre DNIProfesor	Matricula Marca Color
LIBRO	TIENDA	USUARIO	MOVIL
ISBN Titulo Autor NumPaginas	CodTienda Domicilio Telefono	Login Password Nombre FechaAlta	CodMovil Modelo PrecioVenta PrecioCoste

Actualmente no existe una anotación estandarizada universalmente para representar los esquemas del modelo ER, cada recurso bibliográfico o programa utiliza variaciones y ampliaciones sobre la propuesta de Peter Chen.

Intentaremos seguir las siguientes indicaciones:

- No utilizaremos acentos, caracteres especiales ni ñ, sólo cifras y letras
- El nombre de las entidades lo escribiremos en mayúsculas y en singular
- Cada atributo lo escribiremos con la primera letra en mayúscula y el resto en minúsculas
- Si un atributo tiene un nombre compuesto utilizaremos cada palabra empezará en mayúscula y el resto en minúscula: TelefonoPersonal, TelefonoEmpresa, ...
- Si el nombre del atributo son siglas, se escribirá íntegramente en mayúsculas: DNI, ISBN, ...

2.2. Las propiedades

Las propiedades de una entidad hacen referencia a sus características o atributos y también a la relación que puede tener la entidad con otras entidades. Podemos clasificar los atributos en:

- Atributo **principal o secundario**
- Atributo **simple** (*no contiene otros atributos*) o **compuesto** (*contiene otros atributos simples*)
- Atributo **obligatorio** (*debe tener un valor*) u **opcional** (*o nulo, su valor puede no existir o de momento se desconoce*)
- Atributo **monovaluado** (*sólo toma un valor*) o **multivaluado** (*puede tomar varios valores*)
- Atributo **derivado** (*su valor puede conseguirse al realizar operaciones sobre los valores de otros atributos*)

2.2.1 Atributo principal y atributo secundario

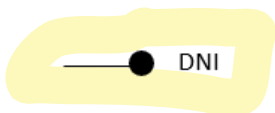
De la colección de atributos que caracteriza a una entidad, uno o más de uno, identifica unívocamente cada uno de sus registros, a este atributo se le denomina **atributo principal**, atributo clave, identificador principal, identificador candidato o **clave primaria**

Por ejemplo, en la entidad **PROFESOR**, el atributo DNI sería el atributo principal, ya que identifica a cada profesor de la entidad **PROFESOR**.

Las características de un atributo principal son:

- Es único para cada registro de la entidad
- Cada registro debe tener un valor en ese atributo
- Este valor es el que se usará para relacionar la entidad con otras entidades
- Todas las entidades deben tener un atributo principal
- El atributo principal puede estar formado por varios atributos simples

La representación de atributo principal depende del modelo utilizado, en el modelo E/R se utiliza



Pero también podemos representarla, poniendo **P.** delante del atributo: **P. DNI** o simplemente subrayando el nombre del atributo DNI.

PROFESOR		ESTUDIANTE		ASIGNATURA		COCHE
<u>DNI</u>		<u>DNI</u>		<u>Nombre</u>		<u>Matricula</u>
Nombre		Nombre		DNIProfesor		Marca
Apellido1		Apellido1				Color
Apellido2		Apellido2				
NumSeguridadSocial		FechaNacimiento				


LIBRO	TIENDA	USUARIO	MOVIL
<u>ISBN</u>	<u>CodTienda</u>	<u>Login</u>	<u>CodMovil</u>
Titulo	Domicilio	Password	Modelo
Autor	Telefono	Nombre	PrecioVenta
NumPaginas		FechaAlta	PrecioCoste

Generalmente un atributo principal está formado por un solo atributo, pero puede suceder que está formado por varios atributos simples:

CUENTABANCARIA
<u>CodBanco</u>
<u>CodSucursal</u>
<u>CodControl</u>
<u>NumCuenta</u>
Tipo
Saldo

En el caso de una **CUENTABANCARIA** el atributo principal está formado por los atributos simples: *CodBanco*, *CodSucursal*, *CodControl*, *NumCuenta*

Algunas entidades tienen **atributos secundarios**, éstos tienen las mismas características que el atributo principal: identifican unívocamente a cada uno de los registros y son obligatorios, pero no se usan para relacionar a la entidad con otra entidad.

En el caso de la entidad **PROFESOR**, el atributo NumSeguridadSocial identifica unívocamente a un profesor, pero no se necesita para relacionar a ese profesor con una asignatura, de eso se encarga el DNI. En el modelo entidad-relación se representa con 

Pero nosotros usaremos la letra **U** con un punto delante del nombre del atributo: **U. NumSeguridadSocial**

EJERCICIO RESUELTO 1

Queremos construir una base de datos que almacene la carta de un restaurante. Para cada plato queremos tener su nombre, descripción, categoría (entrante, plato principal, ...) , foto, precio. Cada plato pertenece a una categoría que se caracteriza por su nombre y breve descripción. Además de los platos, se quiere conocer las recetas para su realización, con la lista de ingredientes necesarios, aportando cantidades, unidad de medida (cl, gr,...) y cantidad actual en el almacén.
Vamos a ver qué información vamos a querer guardar, distinguiendo las entidades y atributos.

Solución:

Primero debemos buscar las entidades, para describir las entidades suelen utilizarse sustantivos, por tanto podríamos tener aquí:

Plato, categorías, recetas, ingredientes

El siguiente paso sería marcar los atributos que describen a cada una de las entidades:

Plato: nombre, descripción, categoría, foto, precio

Categoría: nombre, descripción

Recetas:

Ingredientes: cantidad, unidad, cantidad actual

Receta no tiene atributos, eso nos hace deducir que receta no tiene que ser entidad o que debemos preguntar para tener más información.

Si el enunciado dijera que quieren conocerse los pasos de la receta, entonces sí podríamos decir que receta sería una entidad con por ejemplo los atributos: paso, descripción.

Como hemos visto en los puntos anteriores, es habitual disponer de varios identificadores candidatos para la misma entidad. Por ello ¿cuál elegir como identificador principal?

- Siempre debemos elegir el candidato que tenga más que ver con el problema que estamos resolviendo. Es decir entre un documento nacional de identidad y un código que sólo se usa en la empresa (código cliente, código de socio, nº de personal,...), hay que elegir la segunda opción. La razón es que seguro que en la empresa se tienen más en consideración este segundo número. Como razón técnica : los códigos internos siempre son más cortos que los generales.
- Realmente en el modelo conceptual la única razón a tener en cuenta es la expuesta en el punto anterior. Sin embargo, podemos ahorrar trabajo cuando hagamos el esquema lógico si ya cumplimos estas reglas (que en realidad son técnicas):
 - Sólo debemos elegir como identificadores principales, a atributos (sea uno o varios cuyos valores son únicos) a fechas, números enteros y textos cortos y de tamaño fijo.
 - De los posibles candidatos observados (que cumplan el punto anterior), elegir el que tenga más relación con el problema que estamos resolviendo. Si no encontramos una diferencia conceptual, entonces elegir el que tenga un tamaño más corto.
 - Si ningún candidato cumple estas reglas, es mejor incluso inventarse un identificador (al final contendrá un número entero diferente para cada elemento de la entidad).

En todo caso, cuando pasemos el modelo E/R a su forma lógica (por ejemplo utilizando el modelo relacional) podemos cambiar de idea respecto al identificador.

2.2.2. Atributos simples y atributos compuestos

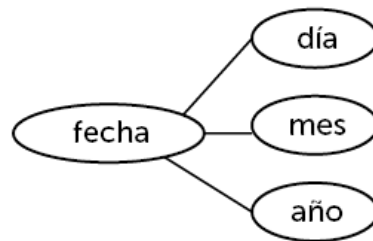
Hasta ahora cuando hemos hablado de atributos, nos estábamos refiriendo a **atributos simples**, que son los que no se pueden subdividir en más atributos. Sin embargo, aquellos que sí pueden dividirse en partes se denominan **atributos compuestos**.

Por ejemplo, el atributo *fechaNacimiento*, podemos dividirlo en día, mes, año. El atributo *cuentaBancaria* formado por 20 dígitos, podríamos dividirlo en banco, sucursal, digControl, numCuenta. El atributo *domicilio* podríamos dividirlo en tipoVia, nombreVia, bloque, piso, puerta, ...

Los atributos compuestos son útiles cuando el usuario a veces hace referencia a ellos como una unidad, pero otras se refiere a cada uno de sus componentes. Si nunca se hace referencia a las partes, es mejor no dividirlo. Para representarlos se utiliza una raya para cada una de las partes. En la representación clásica los atributos que son parte del atributo compuesto se tabulan con respecto al nombre compuesto para indicar que son partes de él:

CUENTABANCARIA
<u>CodCuentaBanco</u>
Cuenta
– CodBanco
– CodSucursal
– CodDC
– NumCuenta
Tipo
Saldo

CUENTABANCARIA
P. CodCuentaBanco
Cuenta
– CodBanco
– CodSucursal
– CodDC
– NumCuenta
Tipo
Saldo



2.2.3. Atributos multivaluados

Generalmente los atributos tienen un solo valor para cada campo de una entidad: atributos **monovaluados**.

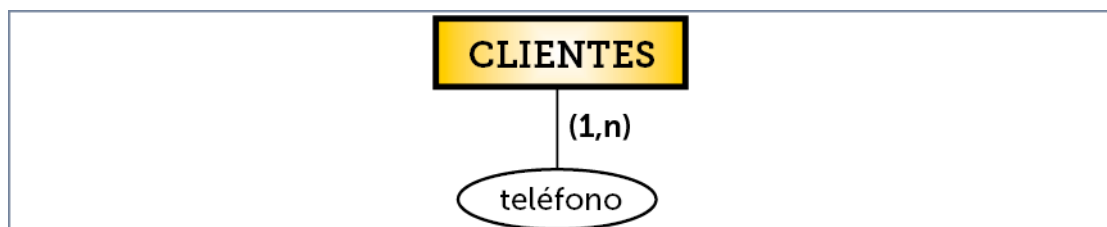
Cuando un atributo puede tomar más de un valor en el mismo campo, hablamos de **atributos multivaluados**.

Estos atributos no cumplen las normas formales de Chen y Codd y por tanto será necesaria su transformación.

A nivel conceptual podemos utilizar atributos multivaluados para simplificar el modelo conceptual que intentamos representar. Por ejemplo atributos como teléfonos, direcciones, idiomas que se habla, etc,...

Para representarlos se utiliza la \rightarrow o **M.** delante del atributo

EMPLEADO		EMPLEADO
<u>DNI</u>		<u>DNI</u>
Nombre		Nombre
Telefono		M. Telefono

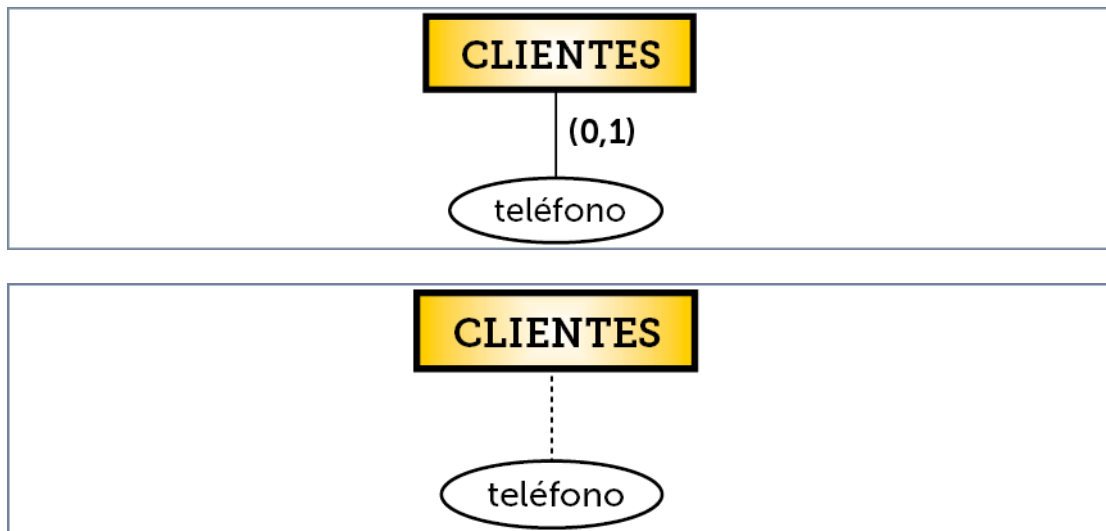


2.2.4. Atributos con valores nulos

Un valor nulo puede indicar la no existencia de un valor para ese atributo o puede representar que aún no se conoce su valor. Por ejemplo al crear una instancia de **PROFESOR**, podemos desconocer el número de teléfono, o por ejemplo puede no tener segundo apellido porque ha nacido en un país en el que sólo se utiliza uno.

Para representarlo se utiliza delante del atributo o **N.** delante del atributo

PROFESOR		ESTUDIANTE		ASIGNATURA		COCHE
<u>DNI</u> Nombre Apellido1 N. Apellido2 NumSeguridadSocial		<u>DNI</u> Nombre Apellido1 N. Apellido2 FechaNacimiento		<u>Nombre</u> N. DNIProfesor		<u>Matricula</u> Marca N. Color
LIBRO		TIENDA		USUARIO		MOVIL
<u>ISBN</u> Titulo Autor N. NumPaginas		<u>CodTienda</u> Domicilio N. Telefono		<u>Login</u> Password N. Nombre FechaAlta		<u>CodMovil</u> Modelo N. PrecioVenta PrecioCoste



2.2.5. Atributos derivados

Son aquellos que se pueden obtener realizando alguna operación con uno o más atributos de una entidad. En realidad no es necesario almacenarlos, a no ser que las operaciones complejas y el almacenamiento optimice el acceso a la información.

Un ejemplo de estos atributos podría ser la letra del DNI o el atributo edad si se dispone de la fechaNacimiento. Se representa con **D.** o **D.** delante del nombre del atributo.

Por ejemplo queremos almacenar los campeonatos de Ajedrez que dirige una asociación, almacenaremos el nombre del campeonato, la fecha de inicio, la fecha de fin y número de días que ha durado (duración). Este último atributo es fácil de calcular conociendo las fechas de inicio y fin:

CAMPEONATO		CAMPEONATO
<u>CodCampeonato</u>		<u>CodCampeonato</u>

Nombre		Nombre
FechaInicio		FechaInicio
FechaFin		FechaFin
□ Duración		D. Duración

EJERCICIO RESUELTO 2

A partir de los siguientes requisitos vamos a ver de qué elementos queremos guardar información, qué entidades y atributos tenemos y de qué tipo:

1. Se quiere construir una base de datos para la cadena de Supermercados Mercatodo que tiene supermercados por toda España. Se quiere guardar información de cada uno de ellos: dirección, metros cuadrados, si está en régimen de alquiler o es comprado y la fecha en la que se efectuó. Se quiere saber quién es el director de cada supermercado. De los directores se almacena: su DNI, su nombre, apellidos, domicilio, teléfono y correo electrónico.
2. Se registra información de los productos que se venden en los distintos supermercados con un código, descripción, familia y género al que pertenece, descuentos para socios, precio e IVA
3. Cuando se realiza una venta se quiere registrar la fecha, los productos que se han vendido, el vendedor que la realiza y, si procede, el cliente al que se vende. El cliente participa cuando se hace una venta a un socio que recibe descuentos especiales. Tanto de los clientes como de los vendedores se quiere almacenar su DNI, Nombre, Apellidos, domicilio, teléfono y correo electrónico.

Solución:

El primer objetivo es localizar los elementos de los que queremos guardar información: las entidades, una vez localizados buscaremos sus características: los atributos.

- Al leer el primer requisito, deducimos que existen las entidades SUPERMERCADO y DIRECTOR. Los atributos de SUPERMERCADO: CodSuper, Direccion, superficie, alquiler y fecha, donde el atributo principal es CodSuper. Para DIRECTOR los atributos son: DNI, Nombre, Apellido1, Apellido2, Domicilio, Telefono y email, donde DNI es el atributo principal
- En el segundo requisito podemos deducir que tenemos la entidad PRODUCTO con los atributos: CodProducto, descripcion, familia, genero, descuentos, precio e IVA.
- Con el tercer requisito vemos que tenemos también la entidad VENTA con los atributos CodVenta y Fecha. También deducimos que tenemos las entidades CLIENTE y VENDEDOR con los atributos DNI, Nombre, Apellido1, Apellido2, domicilio, teléfono, email, donde DNI será el atributo principal.

SUPERMERCADO		DIRECTOR		PRODUCTO
<u>CodSuper</u> Direccion Superficie Alquiler Fecha		<u>DNI</u> Nombre Apellido1 N. Apellido2 Domicilio Teléfono N. Email		<u>CodProducto</u> Descripción Familia Genero N. Descuentos Precio IVA
VENTA		CLIENTE		VENDEDOR
<u>codVenta</u> Fecha		<u>DNI</u> Nombre Apellido1 N. Apellido2 Domicilio Teléfono N. Email		<u>DNI</u> Nombre Apellido1 N. Apellido2 Domicilio Teléfono N. Email

EJERCICIO PROPUESTO 1

Vamos a ampliar los requisitos del ejercicio anterior. Detecta cuáles son los elementos de los que se quiere guardar información, cuáles son las entidades, cuáles los atributos y de qué tipo.

1. Los supermercados, de forma independiente hacen los pedidos a los proveedores con que se relacionan. De los proveedores se quiere conocer su nombre comercial, razón social, dirección social, si es posible la fecha de fundación y el nombre completo del presidente
2. Para personalizar los pedidos se quiere conocer de los proveedores los nombres completos de los empleados con los que se suele trabajar, el cargo que ocupan en la empresa proveedora y el teléfono de contacto
3. Cuando se formaliza un pedido, se registra el empleado que lo gestiona, los productos que se piden y el supermercado que lo hace. En los pedidos se registra la fecha en la que se formaliza.