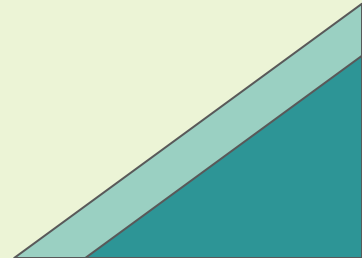


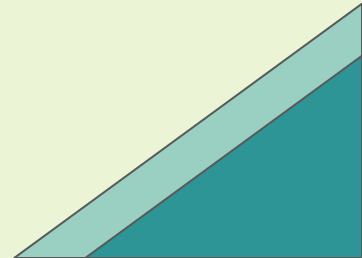
# Conectores y Diagramas de Bases de Datos

Grupo 2



# Index

1. **Quién necesita una base de datos**
2. **Que es un conector**
3. **Que es un diagrama**
  - a. **Diagrama entidad-relación**
  - b. **Modelo relacional**
  - c. **Modelo orientado a objetos**
  - d. **Modelo lógico de bases de datos**
4. **Modelo físico de bases de datos**
5. **Qué tipos de conector tengo**
6. **Cuando se utiliza un conector**
7. **Ejemplos de cómo utilizar un diagrama de bases de datos**
8. **Diagrama ER caso practico: Fruteria**



# 1. Quién necesita una base de datos

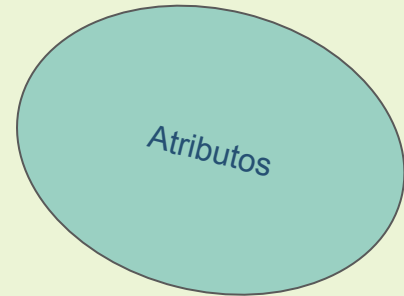
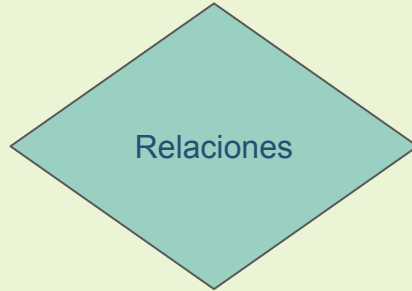
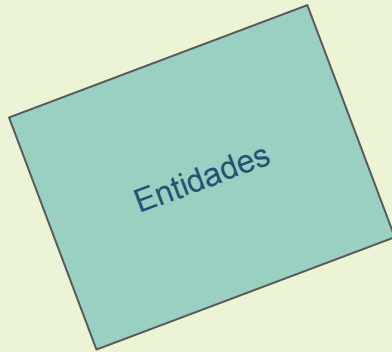
Las entidades que existen en nuestro universo de discurso, por ejemplo: Tenemos una tienda de informática que necesita una base de datos necesitaremos entidades como Clientes, Artículos, Empleados, Sucursal etc ...

Las relaciones entre las diferentes entidades.



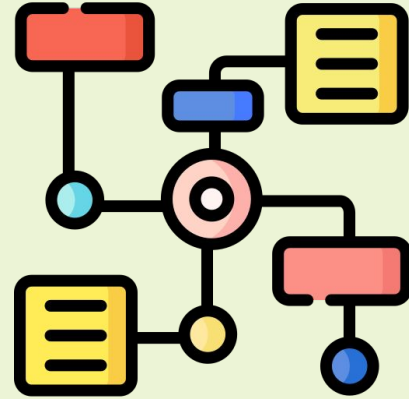
## 2. Qué son los Conectores

Es un estándar de acceso a las BBDD con objetivo de hacer posible el acceso a cualquier dato desde cualquier aplicación sin importar el gestor de base de datos, utilizando ciertas figuras con diferentes funciones.



### 3. Que es un diagrama de bases de datos

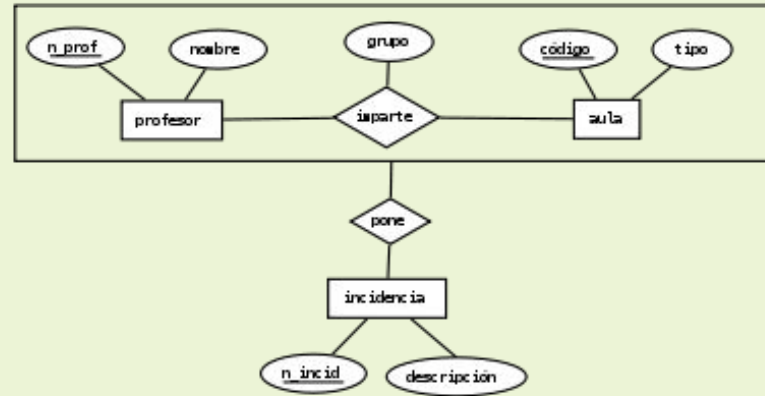
Un diagrama de bases de datos es una representación visual de una base de datos. Utiliza símbolos estándar para representar tablas, columnas, relaciones y otras entidades de base de datos. Los diagramas de bases de datos son una herramienta importante para el diseño, desarrollo y mantenimiento de bases de datos.



## 3.1 Diagrama entidad relación

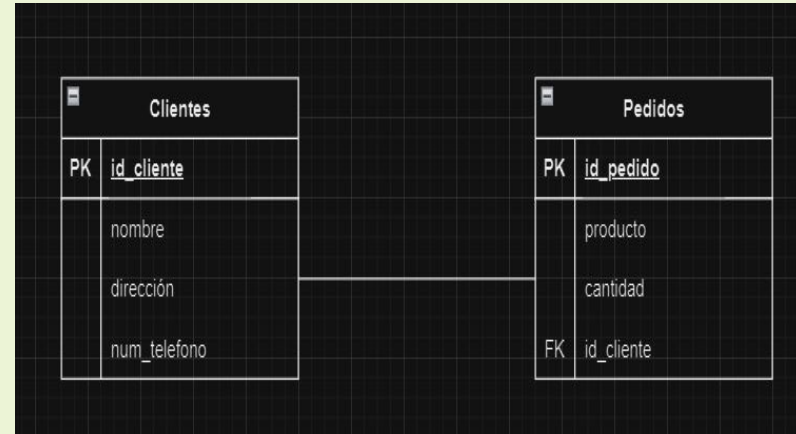
Descripción: Representa visualmente la estructura de una base de datos mostrando las entidades y las relaciones entre ellas.

También conocidos como los ERD o modelos ER, emplean un conjunto definido de símbolos, tales como rectángulos, diamantes, óvalos y líneas de conexión para representar la interconexión de entidades, relaciones y sus atributos.



## 3.2 Diagrama relacional

El modelo relacional ordena los datos en tablas y constan de campos divididos en filas y columnas. Se pueden establecer relaciones entre los datos de las tablas, mejorando la funcionalidad de la gestión de los datos.



## 3.2 Diagrama relacional caso práctico:FRUTERIA

Ejercicio 2.- En un comercio, un cliente compra varios pdtos., y un pdto puede ser comprado por varios clientes. Al cliente se le asigna un código para identificarle, también interesa conocer su nombre, dirección, edad y teléfono. El pdto también tiene un código para identificarle. Se quiere almacenar el nombre o descripción del pdto y el precio por unidad. También se pretende almacenar la cantidad de pdto que compra cada cliente.



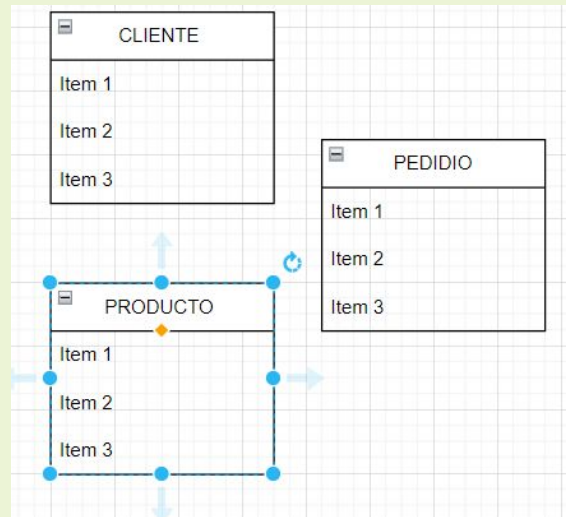


## 3.2 IDENTIFIQUE LOS OBJETOS DE DATOS PRINCIPALES

Los objetos de datos principales son los conceptos o entidades que se almacenarán en la base de datos. Por ejemplo, si está creando una base de datos para una tienda minorista, los objetos de datos principales podrían incluir productos, clientes y pedidos.

### OBJETOS PRINCIPALES:

- CLIENTE
- PRODUCTO
- PEDIDO



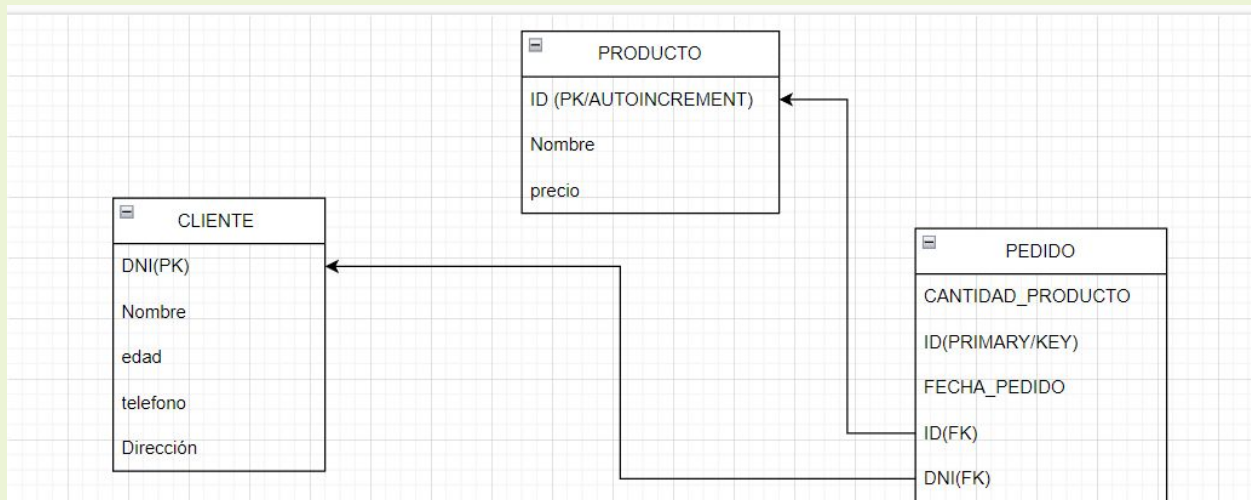
## 3.2 DEFINA LOS ATRIBUTOS DE LOS OBJETOS DE DATOS

Los atributos son los datos que describen cada objeto de datos. Por ejemplo, el objeto de datos "producto" podría tener atributos como el nombre del producto, el precio y la descripción.



## 3.2 DEFINA LOS ATRIBUTOS DE LOS OBJETOS DE DATOS

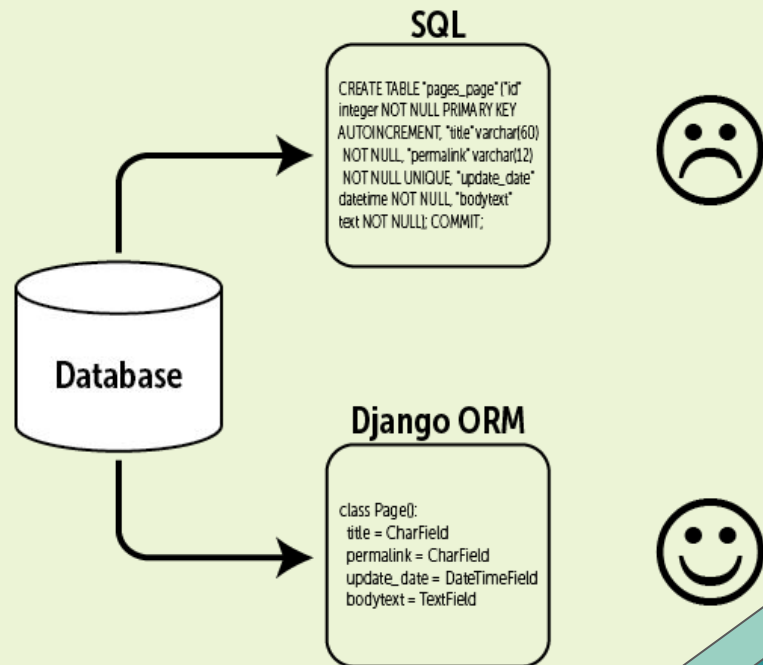
Las relaciones son los vínculos entre los objetos de datos. Por ejemplo, un producto podría estar relacionado con un cliente si el producto fue comprado por ese cliente



## 3.3 Diagrama orientado a objetos

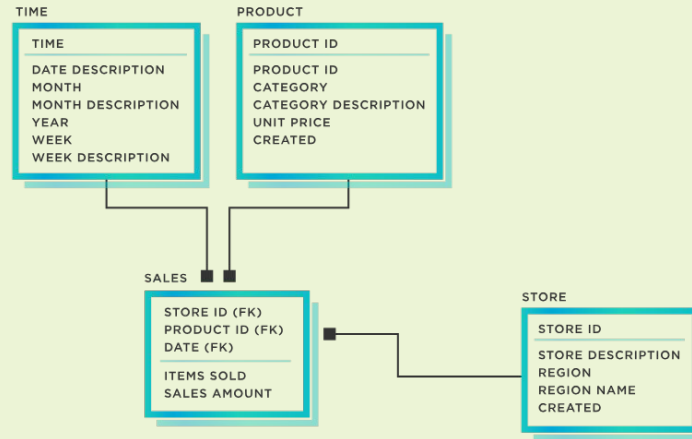
Normalmente cuando interactuamos con una base de datos no utilizamos comandos de SQL para crear tablas, hacer queries etc ... Lo que se utiliza en frameworks de back office como Flask, Django, Spring son ORM que significa **Object Relational Mapping**.

Básicamente es un nivel de abstracción que ofrece una librería para manipular datos en una base de datos que suele ser creando nosotros nuestras propias clases heredando de clases de la librería.



## 3.4 Modelo lógico de bases de datos

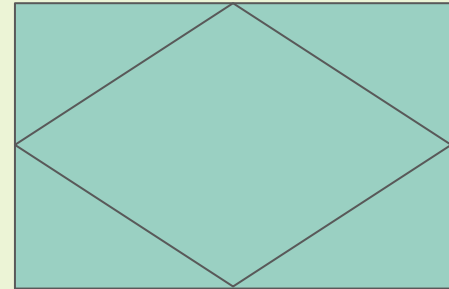
Un modelo lógico de datos (LDM) proporciona las especificaciones de datos que describen los conceptos, las relaciones y la interpretación de los valores de los datos.



## 4. Qué tipos de conector tengo?

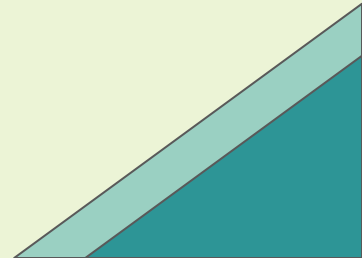
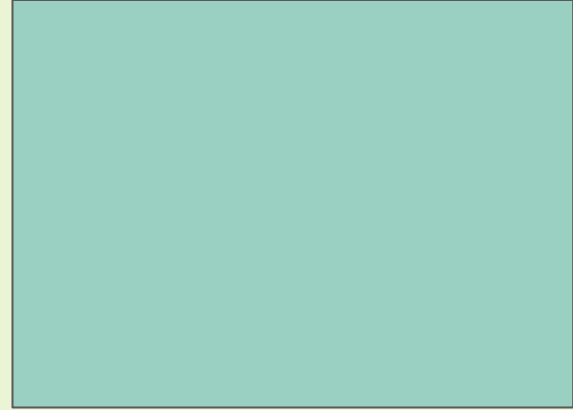
### Símbolos de entidades de diagramas ER:

Las entidades son objetos o conceptos que representan datos importantes, es decir, son generalmente sustantivos como producto, cliente, ubicación etc. Hay 3 tipos de entidades utilizados en los diagramas de entidad-relación. Se representan con un rectángulo.



# Entidad Fuerte

Estas figuras son independientes de otras entidades y con frecuencia se les denomina entidades matriz ya que a menudo tienen entidades débiles que dependen de ellas. También tendrán una clave primaria, que distinga a cada suceso de la entidad.



# Entidad debil

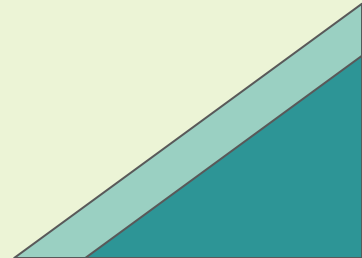
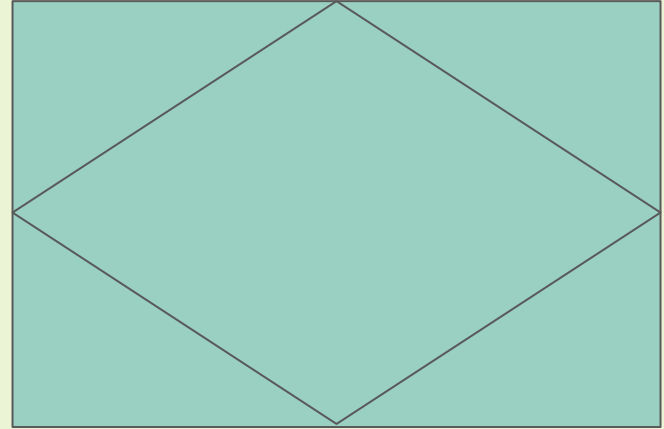


Las entidades débiles dependen de algún otro tipo de entidad. No tienen claves primarias y no tienen significado en el diagrama sin su entidad matriz.



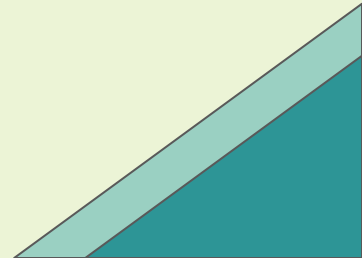
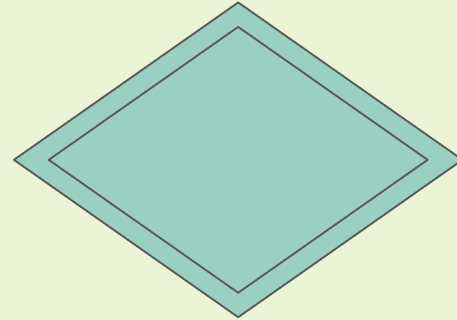
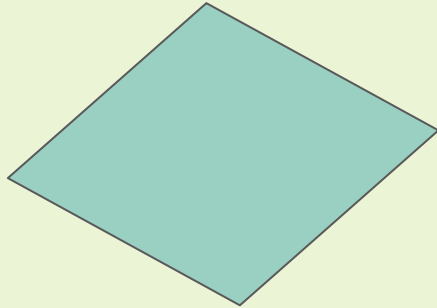
# Entidad asociativa

Las entidades asociativas relacionan las instancias de varios tipos de entidades. También contienen atributos que son específicos a la relación entre esas instancias de entidades.



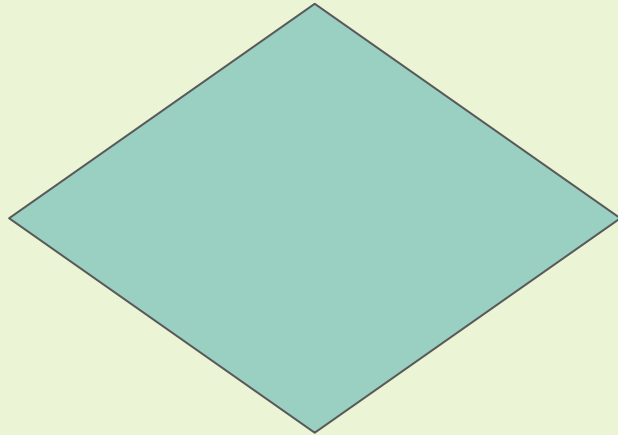
# Símbolos de relaciones en diagramas ER

Las relaciones se usan para documentar la interacción entre dos entidades, es decir, son generalmente son verbos como asignar, asociar o rastrear y proporcionar información útil. Se representa como un diamante.



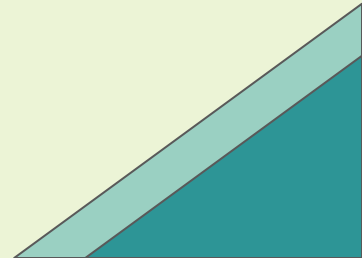
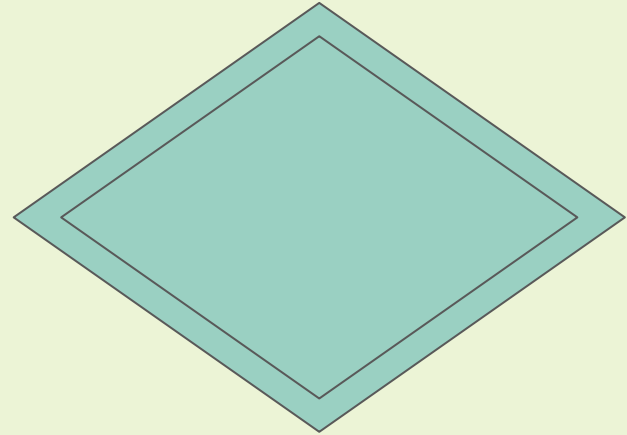
# Relación

Las relaciones son asociaciones entre dos o más entidades.



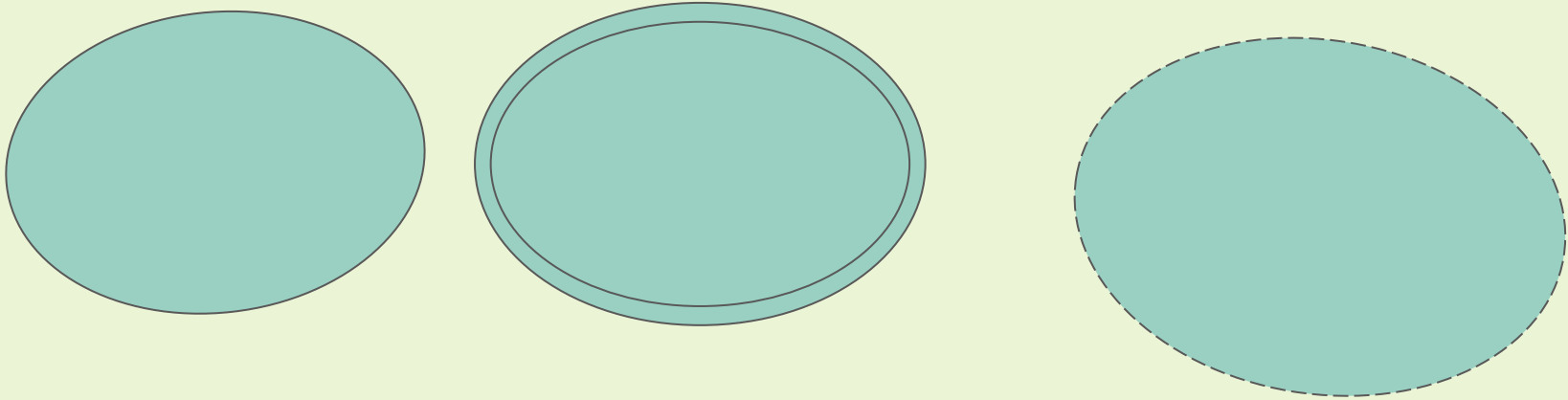
# Relación débil

Las relaciones débiles son conexiones entre una entidad débil y su propietario.

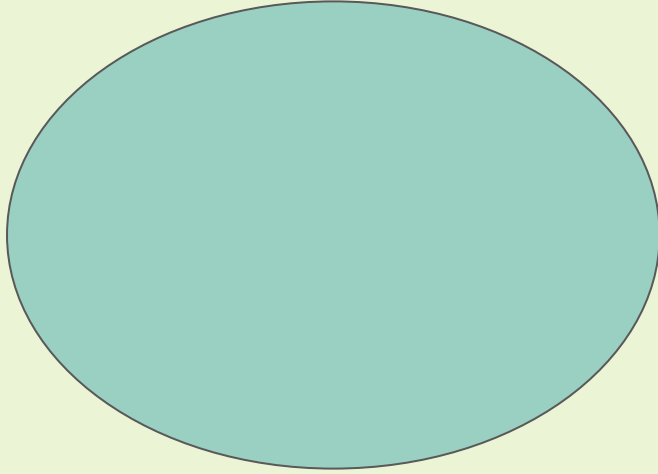


# Símbolos de atributos de diagramas ER

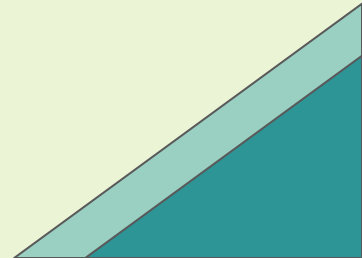
Los atributos son las características de la entidad que ayudan a los usuarios a entender mejor la base de datos. Los atributos se incluyen para agregar detalles de las diversas entidades que hay en el diagrama.



# Atributo

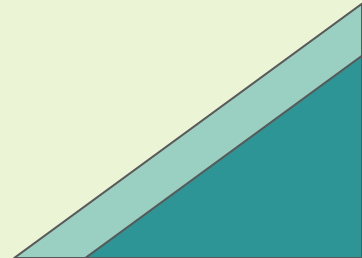
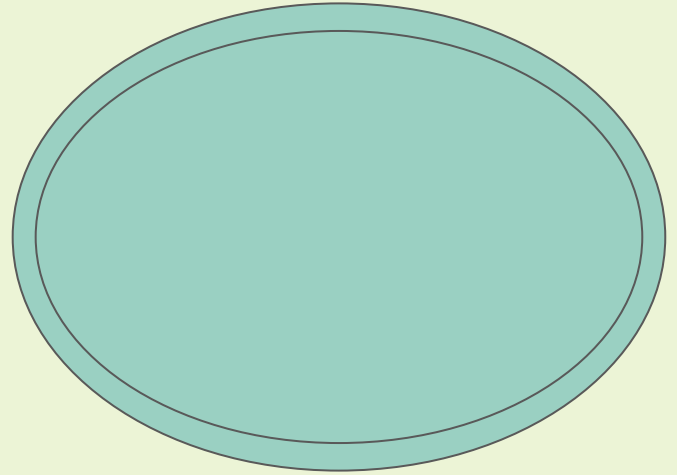


Los atributos son las características de una entidad, una relación de muchos a muchos, o una relación de uno a uno.

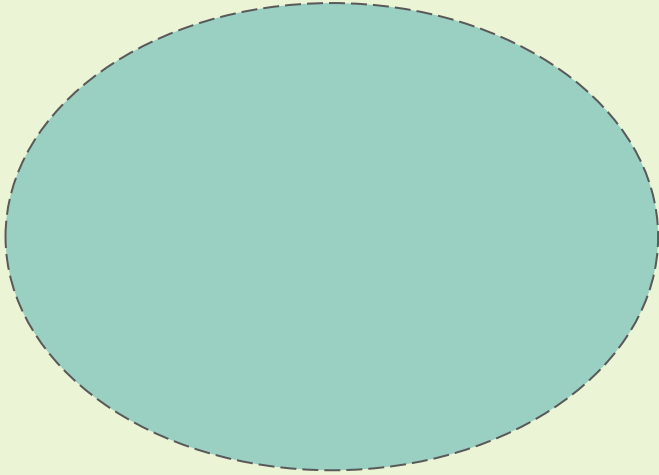


# Atributo de varios valores

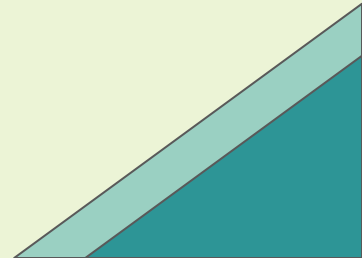
Los atributos de valores múltiples son aquellos que pueden tomar más de un valor.



# Atributo derivado

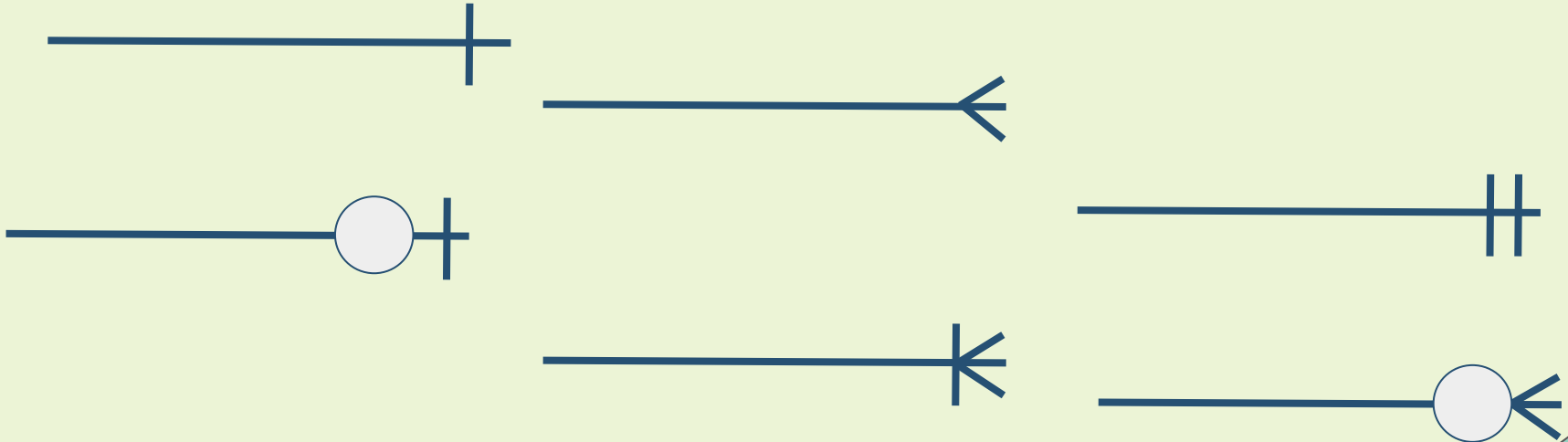


Los atributos derivados son atributos cuyos valores se pueden calcular a partir de valores de atributos relacionados.



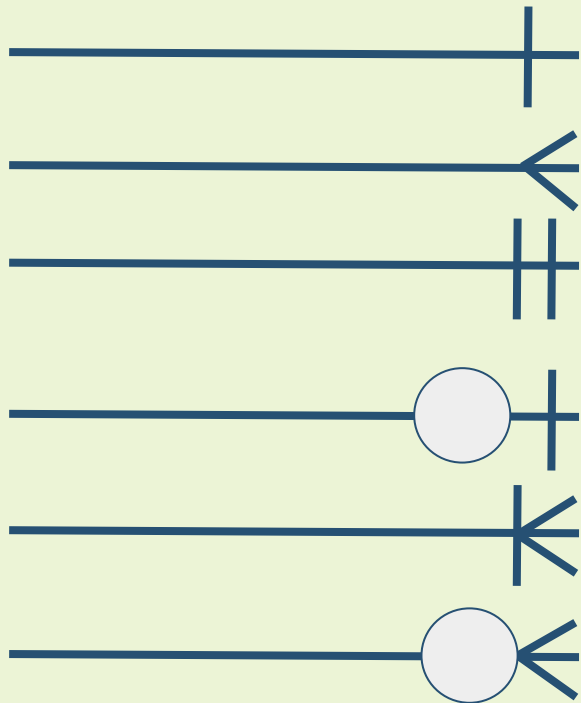
# Cardinalidad y ordinalidad

Cardinalidad se refiere al número máximo de veces que una instancia en una entidad se puede relacionar con instancias de otra entidad. Por otra parte, ordinalidad es el número mínimo de veces que una instancia en una entidad se puede asociar con una instancia en la entidad relacionada.





# Ejemplos



La cardinalidad y la ordinalidad se muestran a través del estilo de una línea y su extremo, según el estilo de notación seleccionado.

## 9.0 Diagrama ER caso practico: Fruteria

Ejercicio: En un comercio, un cliente compra varios pdtos y un pdto puede ser comprado por varios clientes. Al cliente se le asigna un código para identificarle, también interesa conocer su nombre, dirección, edad y teléfono. El pdto también tiene un código para identificarle. Se quiere almacenar el nombre o descripción del pdto y el precio por unidad.

También se pretende almacenar la cantidad de pdto que compra cada cliente.

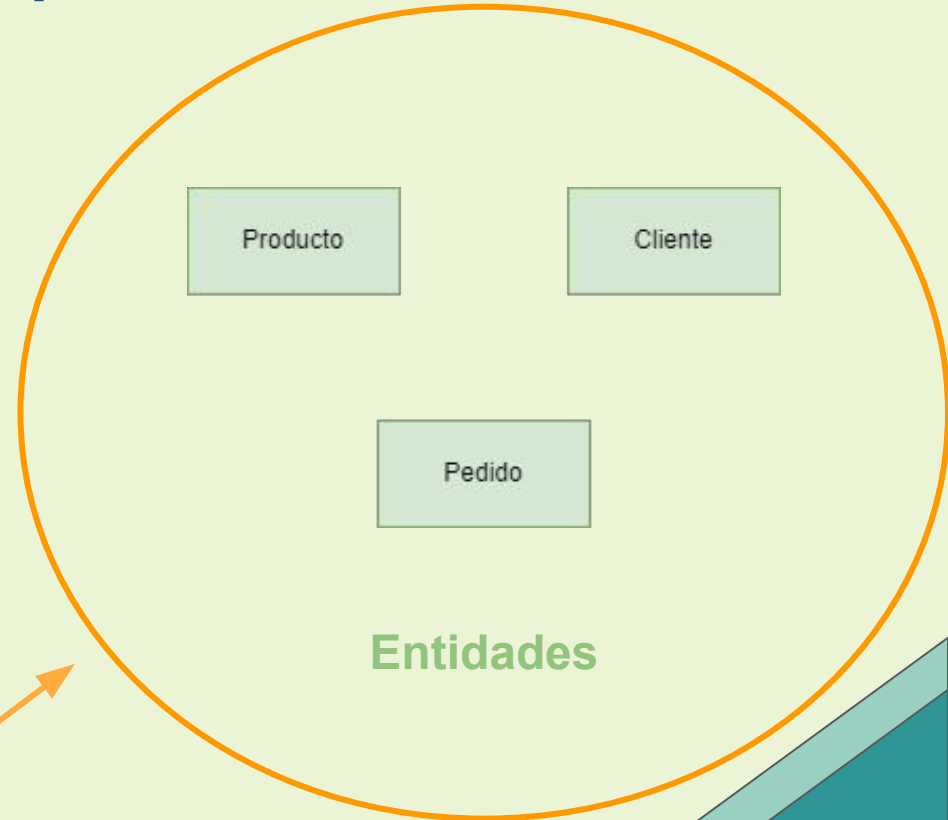


## 9.1 Identificamos entidades que necesitamos

Ejercicio: En un **comercio**, un cliente compra varios productos y un **producto** puede ser comprado por varios clientes. Al **cliente** se le asigna un código para identificarle, también interesa conocer su nombre, dirección, edad y teléfono. El pdto también tiene un código para identificarle. Se quiere almacenar el nombre o descripción del pdto y el precio por unidad.

También se pretende almacenar la cantidad de pdto que compra cada cliente.

Para almacenar la cantidad de producto que comprará un cliente crearemos una entidad **Pedido** para hacer de intermediario.

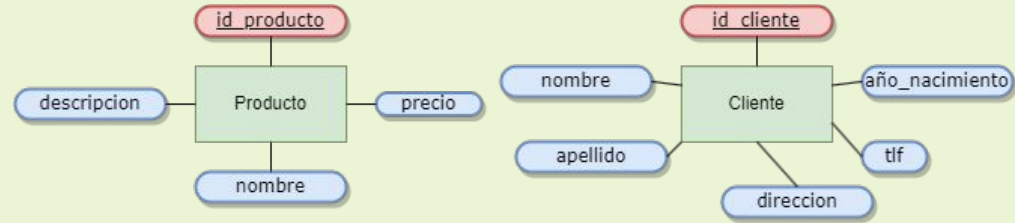


**Universo de discurso: Frutería**

## 9.2 Añadimos atributos a nuestras entidades

**PKI:** Las entidades producto, cliente y pedido necesitan un identificador único esto será un **atributo clave** tenemos que subrayar el texto.

**Atributos:** Para la entidad cliente necesitaremos crear **nombre**, **apellido**, **dirección**, **año de nacimiento** y **teléfono** y para la entidad producto **nombre**, **descripción** del pdto y el **precio** por unidad. Luego para la entidad pedido añadiremos un atributo **fecha pedido**.



Atributos clave

Atributos

## 9.3 Añadimos atributos Derivados

**Atributos derivados:** Crearemos los atributos llamados **edad** y **nombre\_completo**, el atributo edad del cliente saldrá del año de nacimiento y el año actual y el nombre completo sera la combinacion de nombre y apellido, se marcará con una **elipse con línea discontinua**.

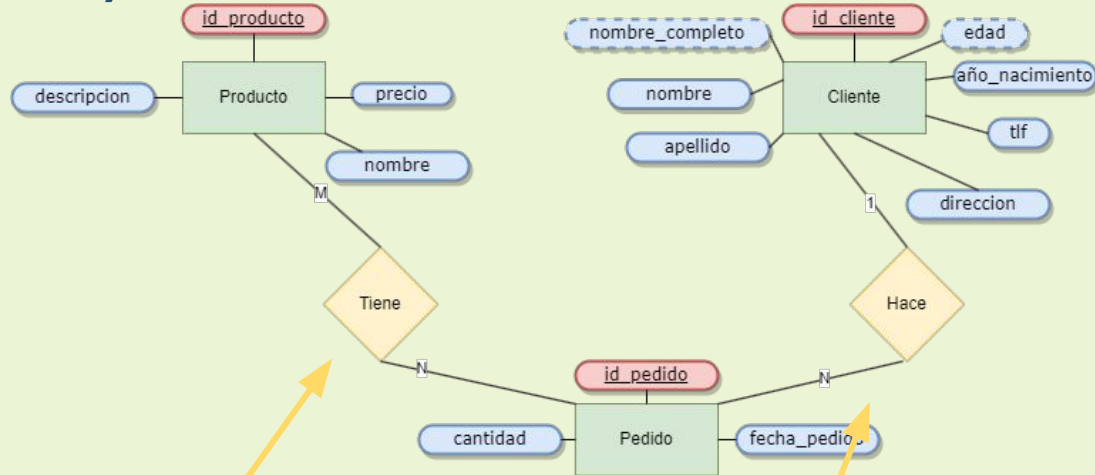


## 9.4 Relación (Cardinalidad)

**Cardinalidad:** El número de instancias que pueden relacionarse **N-M**, **1-1** y **1-N**.

**Relación cliente-pedido:** Un cliente puede **hacer muchos pedidos** o **0** pero **1 pedido SOLO TENDRÁ 1** cliente, sería **1-N 1-muchos**.

**Relación producto-pedido:** Un producto puede tener **0 o muchos pedidos** y un pedido puede tener **0 o muchos productos**, sería una relación **N-M**.



**Muchos-Muchos**

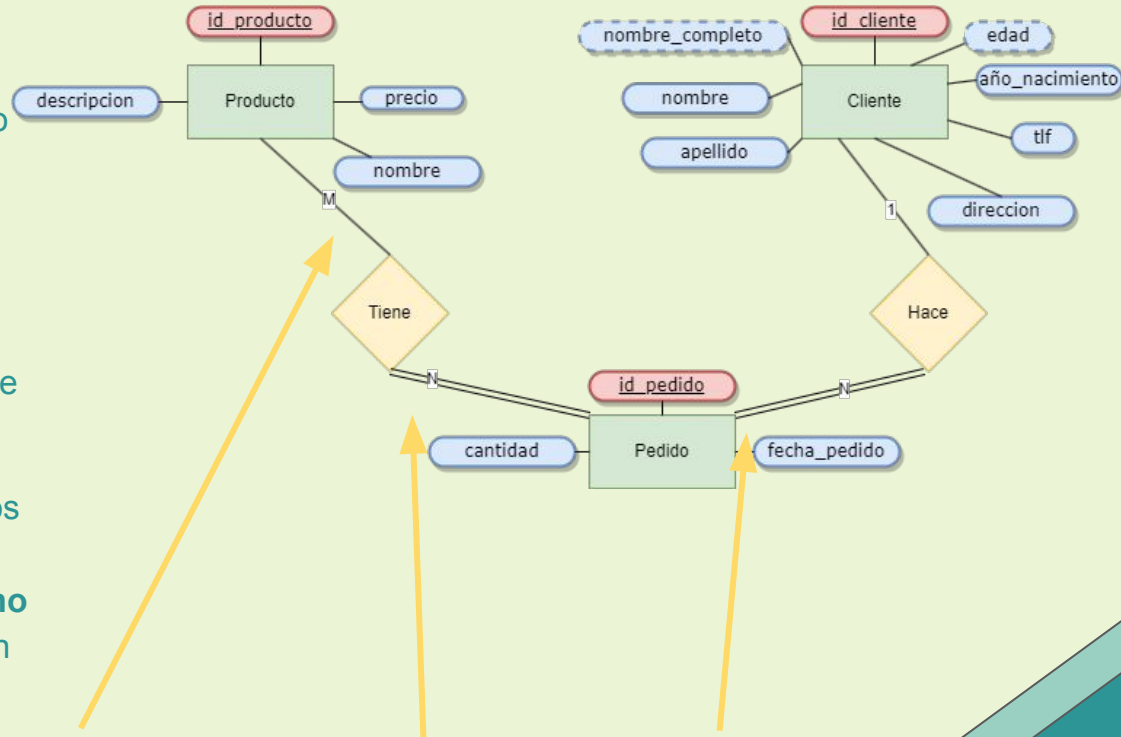
**1-Muchos**

## 9.5 Restricciones (Participación)

**Participación:** Indica si todas o no todas las entidades participan en la relación, sería como cardinalidad mínima.

**Participación cliente-pedido:** Todos los pedidos **deben tener un cliente**, ya que no tendría sentido que existieran sin uno. Sin embargo, no todos los clientes necesariamente **deben realizar un pedido**.

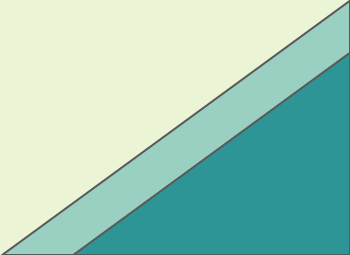
**Relación producto-pedido:** Todos los pedidos deben incluir **al menos un producto**, de lo contrario, carecería de sentido. No obstante, **no todos los productos necesitan participar** en un pedido.



Participacion parcial

Participacion total

# Quién explicado que:

- Quién necesita una base de datos (robert)
  - Que es un conector(Austin)
  - Que es un diagrama(oscar)
  - Diagrama entidad relación(Angels)
  - Modelo relacional (santi)
  - Modelo orientado a objetos (robert)
  - Modelo lógico de bases de datos (jesús)
  - Qué tipos de conector tengo(Austin)
  - Cuando se utiliza un conector (Oscar)
  - Diagrama ER caso práctico: Frutería (robert)
  - Diagrama crowfoot caso práctico: Frutería (santi)
- 



# Bibliografía

<https://www.lucidchart.com/pages/es/que-es-un-modelo-de-base-de-datos>

<https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>

[database - What is an ORM, how does it work, and how should I use one? - Stack Overflow](#)

<https://www.lucidchart.com/pages/es/simbolos-de-diagramas-entidad-relacion>

<https://www.oracle.com/es/database/what-is-a-relational-database/>

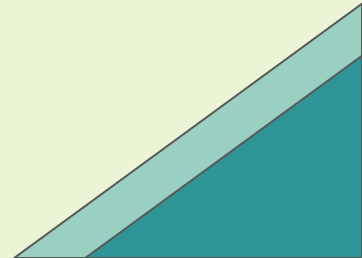
<https://keepcoding.io/blog/modelo-de-datos-fisicos/>

<https://www.quora.com/What-are-some-benefits-and-limitations-of-logical-data-modelling>

<https://www.sciencedirect.com/science/article/abs/pii/B9780128002056000044>

<https://www.sciencedirect.com/science/article/abs/pii/B9780123851260000115>

<https://www.tibco.com/reference-center/what-is-a-logical-data-model>



# Conectores y Diagramas de Bases de Datos

Gracias por vuestra atención

