

UNIVERSIDAD
AUSTRAL



SOMOSAUSTRAL

El mundo de las bases de datos

Un pantallazo





Introducción

Definiciones

f. Inform. Conjunto de datos organizados de tal modo que permita obtener con rapidez diversos tipos de información

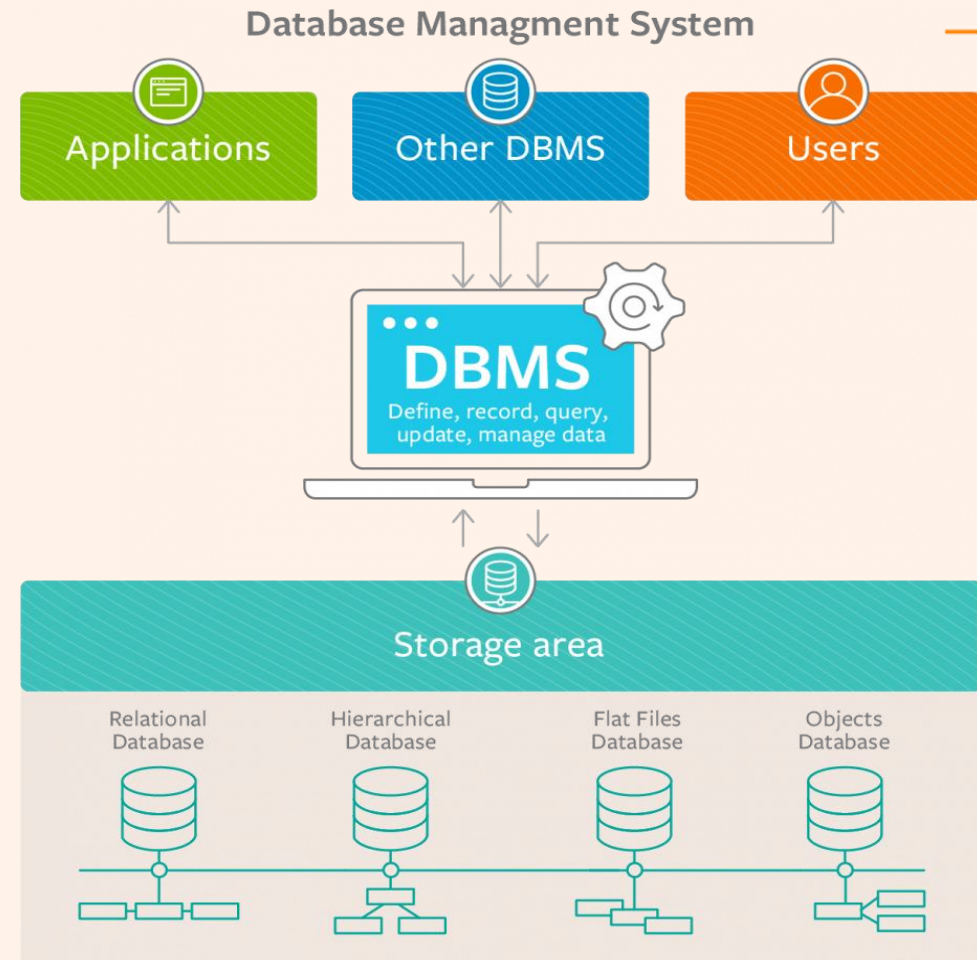
RAE

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Introducción

Sistema de Gestión de Bases de Datos

Un sistema de gestión de bases de datos (Database management system - DBMS) es un conjunto de programas que se utilizan para definir, administrar y procesar bases de datos y sus aplicaciones asociadas.





Introducción

Capacidades del DBMS



Consulta y actualización de los datos



Mantenimiento de esquemas



Almacenamiento persistente



Manejo de transacciones









Interface de programación



Introducción

Beneficios del DBMS

-  Seguridad de datos
-  Compartir datos
-  Acceso a datos y auditoría
-  Integración de datos
-  Abstracción e independencia
-  Gestión y administración

RDBMS

Sistema de gestión de bases de datos relacionales

Modelo relacional

Una base de datos relacional es una base de datos basada en el modelo relacional (relational model), propuesto por E. F. Codd en 1970. Un sistema de software utilizado para mantener las bases de datos relacionales es un sistema de gestión de bases de datos relacionales (RDBMS).



Modelo Entidad-Relación



Introducción

Modelo Entidad-Relación

- El proceso de diseño de una base de datos comienza con un análisis de qué información debe contener la base de datos y cuál es la relación entre los componentes de la información.
- La estructura de la base de datos, llamada esquema de la base de datos, se especifica en uno de varios lenguajes o notaciones adecuadas para expresar diseños.
- Después de la debida consideración, el diseño se compromete a una forma en la que se puede ingresar en el DBMS, y la base de datos adquiere existencia física.



Elementos

Modelo Entidad-Relación

- **Conjuntos de entidades:** una entidad es un objeto abstracto de algún tipo, y una colección de entidades similares forma un conjunto de entidades.
- **Atributos:** un conjunto de entidades tiene atributos asociados, que son propiedades de las entidades de ese conjunto.
- **Relaciones:** las relaciones son conexiones entre dos o más conjuntos de entidades.
- **Diagrama entidad-relación:** un diagrama E / R es un gráfico que representa conjuntos de entidades, atributos y relaciones.

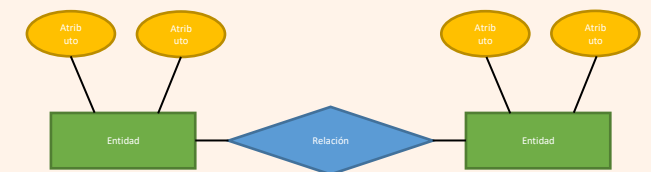
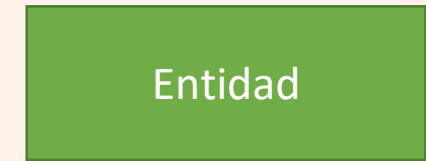


Diagrama E/R

Modelo Entidad-Relación

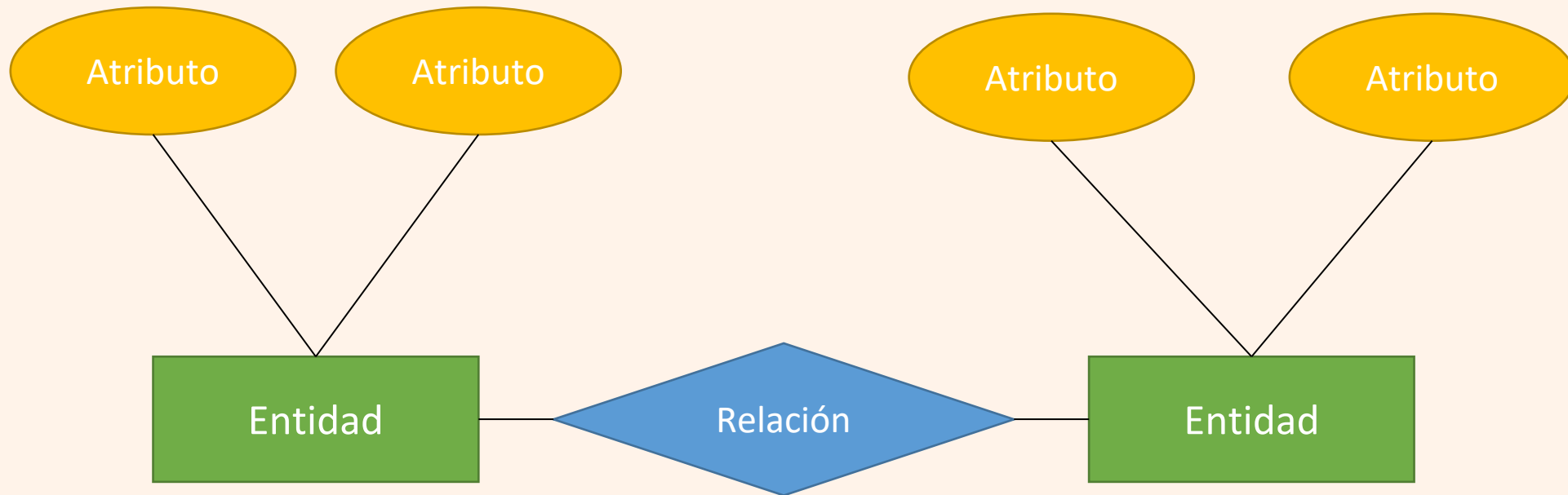
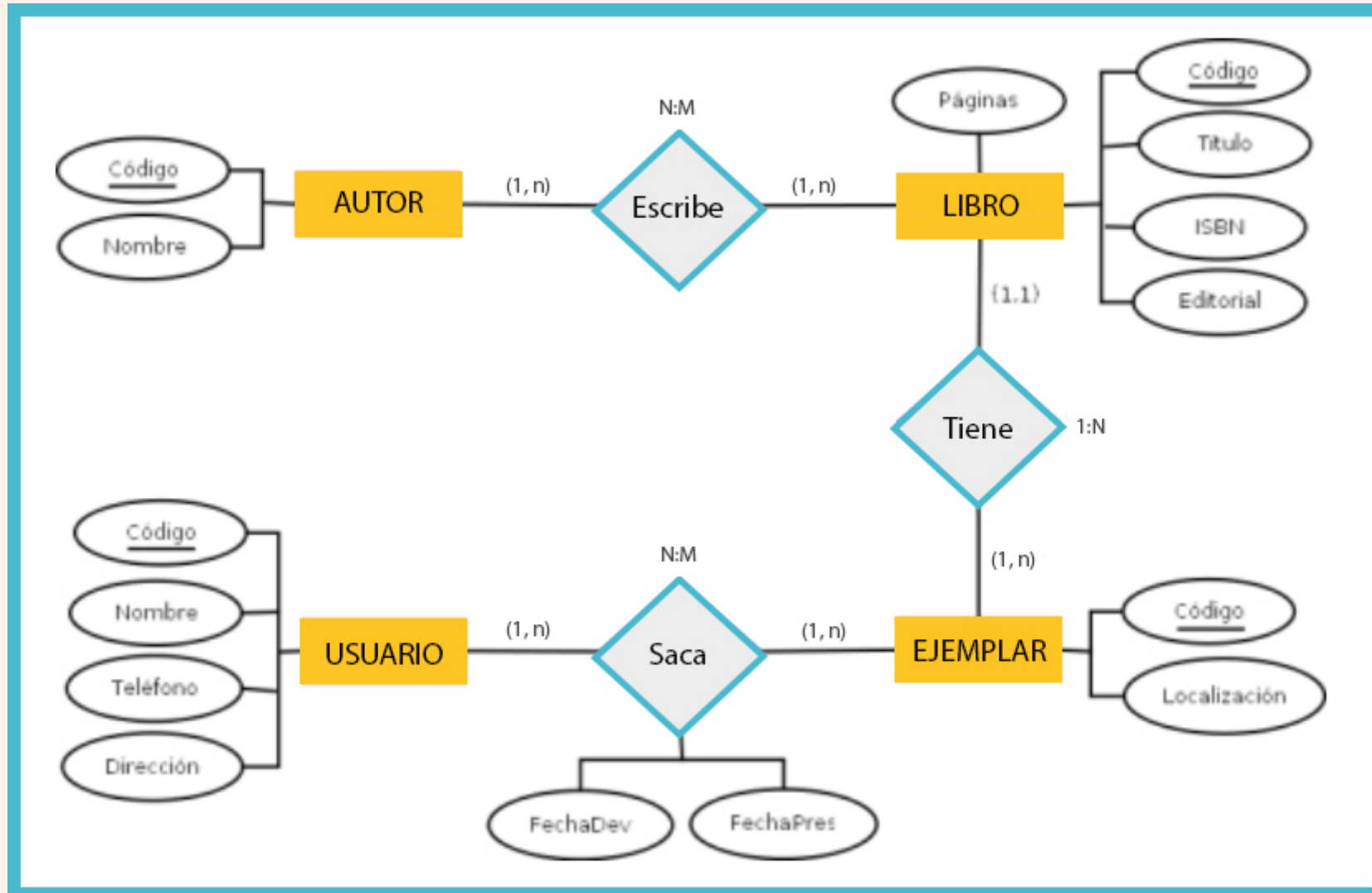


Diagrama E/R

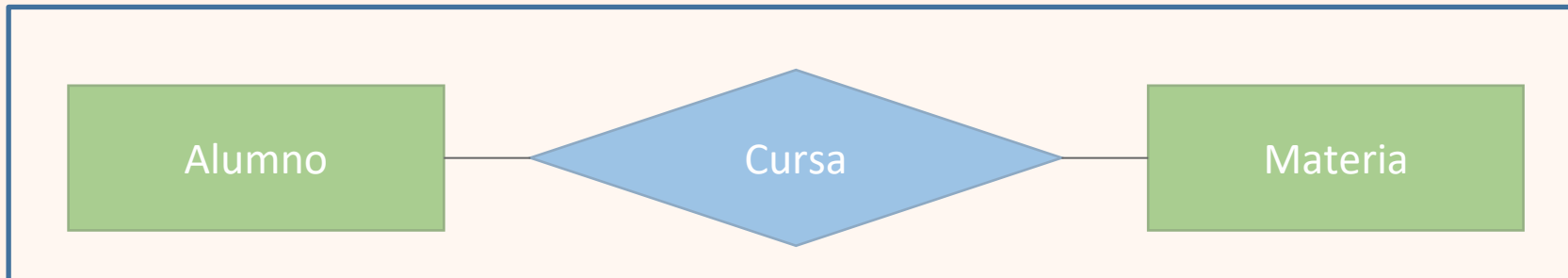
Modelo Entidad-Relación



Relación Binaria

Modelo Entidad-Relación

Una relación binaria puede conectar cualquier miembro de un conjunto de entidades con cualquier número de miembros de otro conjunto de entidades. Sin embargo, es común que exista una restricción sobre la "multiplicidad" de una relación.

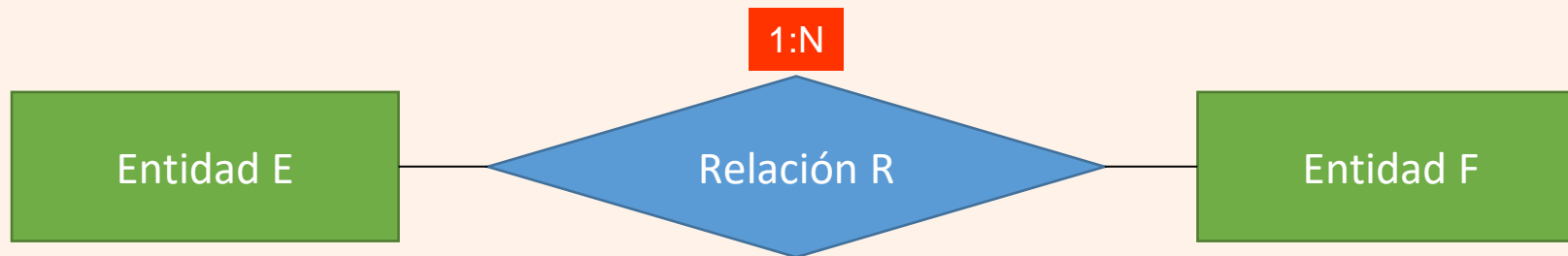


Cardinalidad de Relación Binaria

Modelo Entidad-Relación

Tipo de Cardinalidad

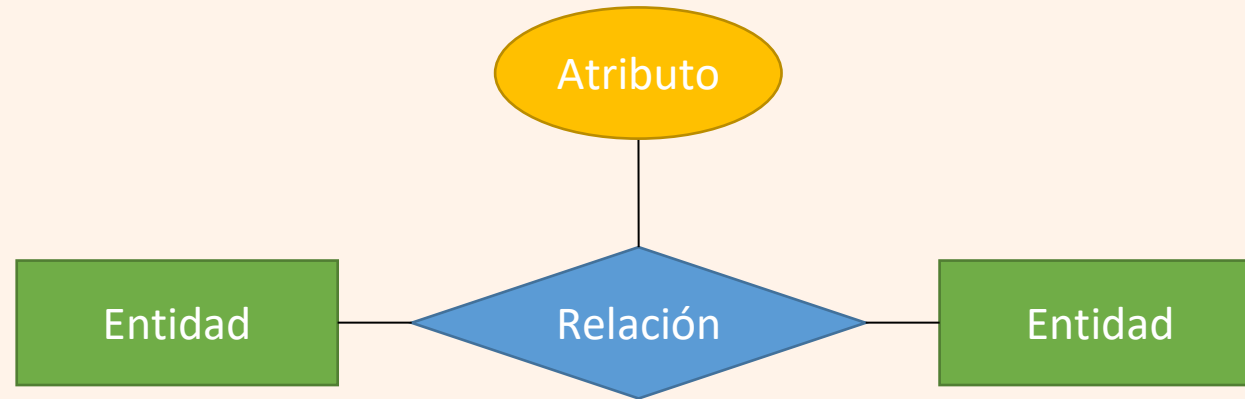
- **UNO a UNO:** Cada miembro de la entidad E está relacionado con un solo miembro de la entidad F, y viceversa.
- **UNO a MUCHOS:** Cada miembro de la entidad E puede estar relacionado con más de un miembro de la entidad F, pero cada miembro de F está relacionado sólo con un miembro de E.
- **MUCHOS A MUCHOS:** Cada miembro de la entidad E puede estar relacionado con más de un miembro de la entidad F, y a su vez cada miembro de F puede estar relacionado con más de un miembro de E.



Atributos en Relaciones

Modelo Entidad-Relación

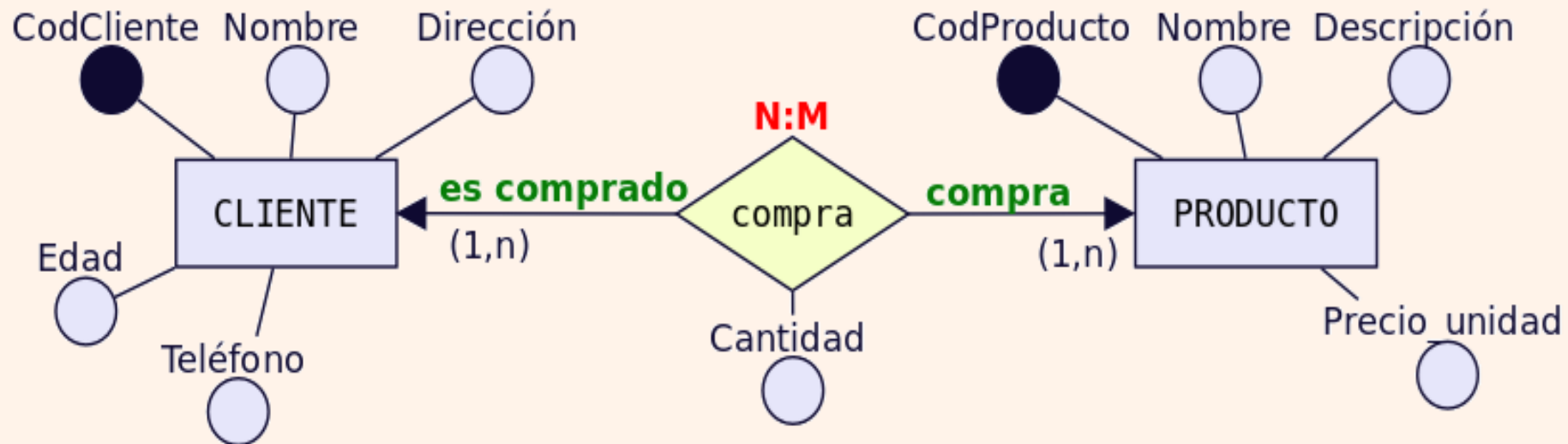
- A veces es conveniente, o incluso esencial, asociar atributos con una relación, en lugar de con cualquiera de los conjuntos de entidades que conecta la relación.



Atributos en Relaciones

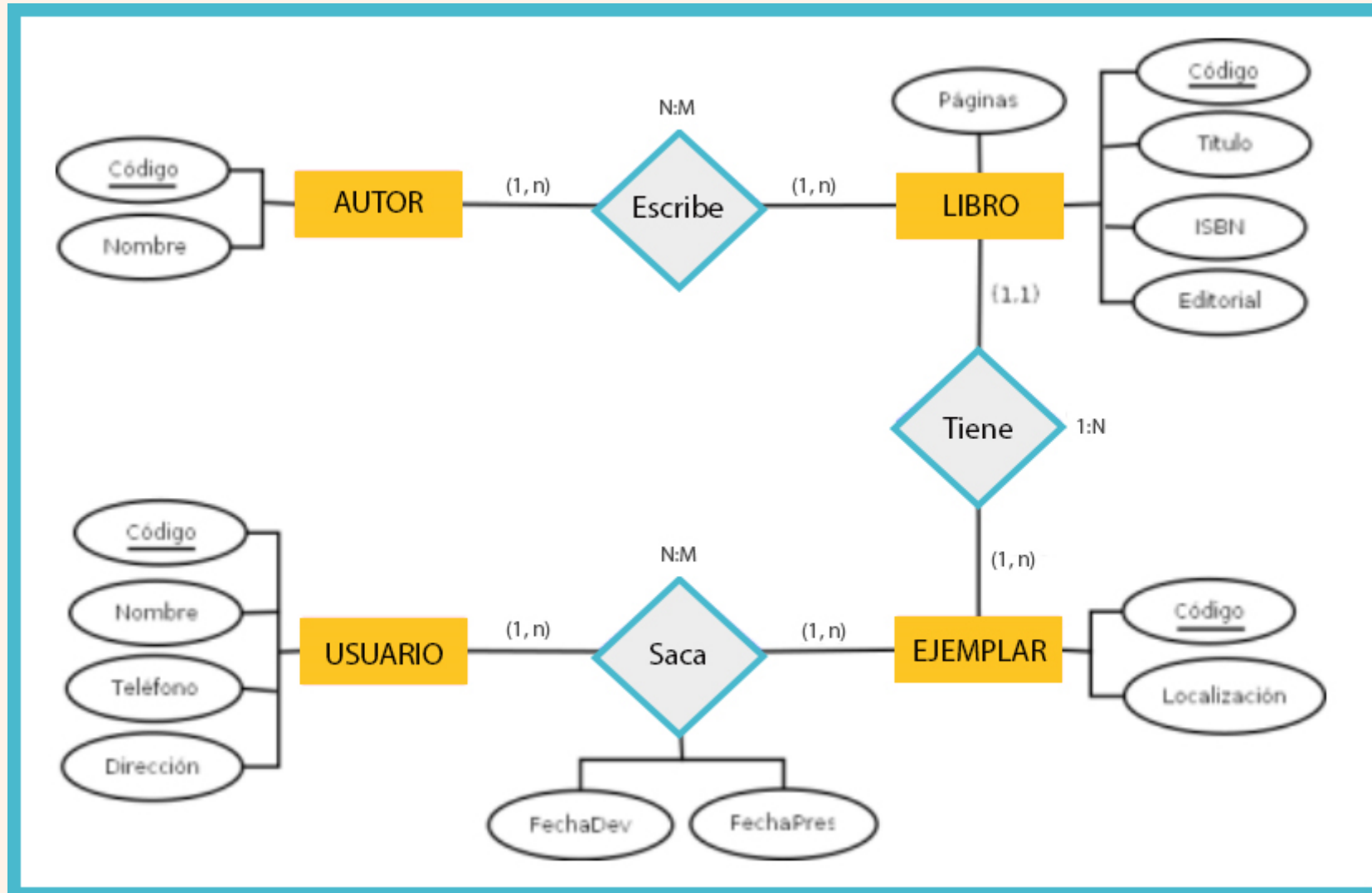
Modelo Entidad-Relación

- A veces es conveniente, o incluso esencial, asociar atributos con una relación, en lugar de con cualquiera de los conjuntos de entidades que conecta la relación.



Atributos en Relaciones

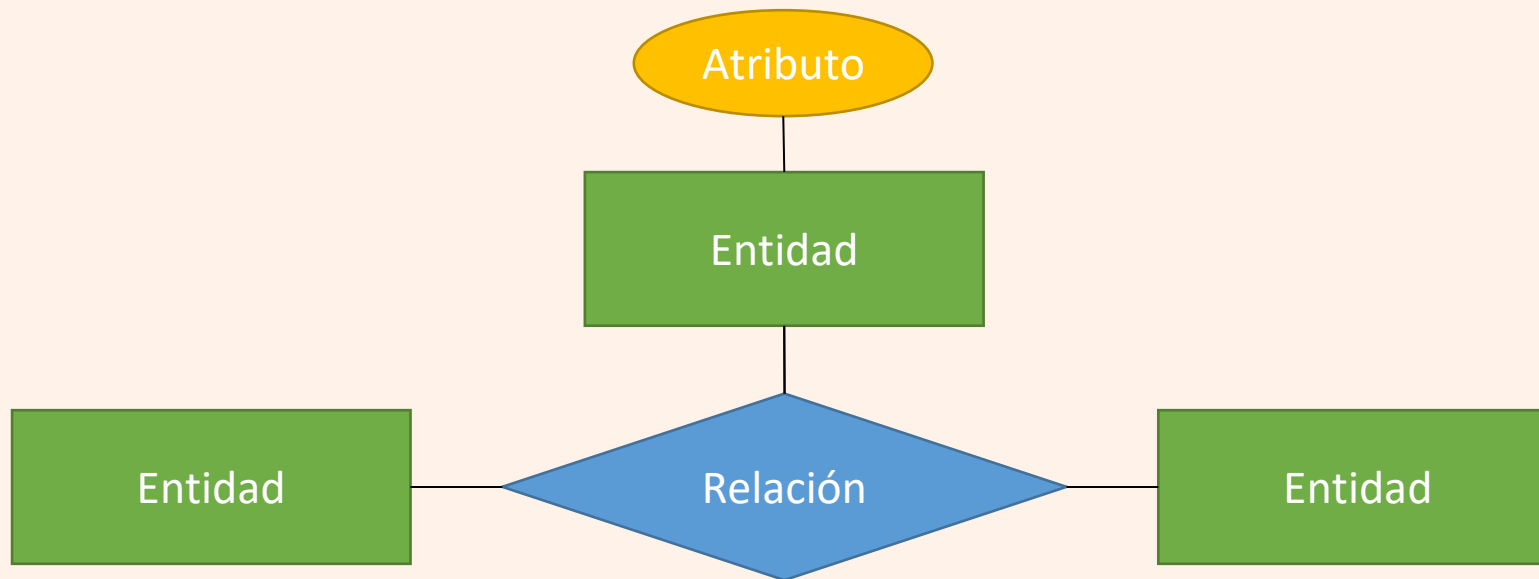
Modelo Entidad-Relación



Atributos en Relaciones

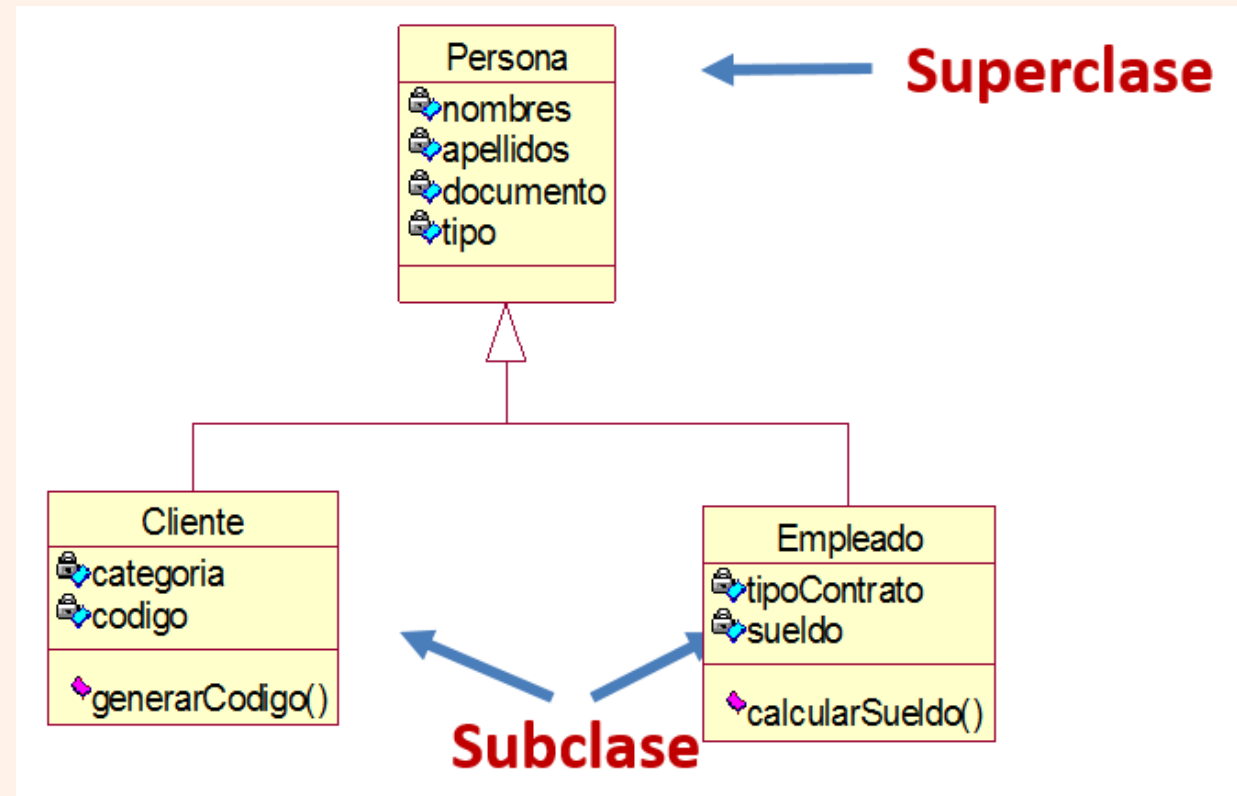
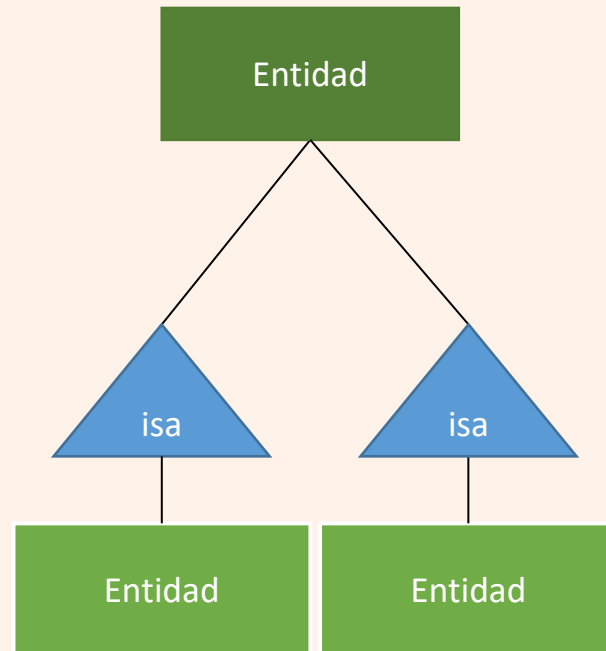
Modelo Entidad-Relación

- No siempre es necesario colocar atributos en las relaciones. En su lugar, podemos inventar un nuevo conjunto de entidades, cuyas entidades tienen los atributos atribuidos a la relación. Si luego incluimos este conjunto de entidades en la relación, podemos omitir los atributos de la relación en sí.



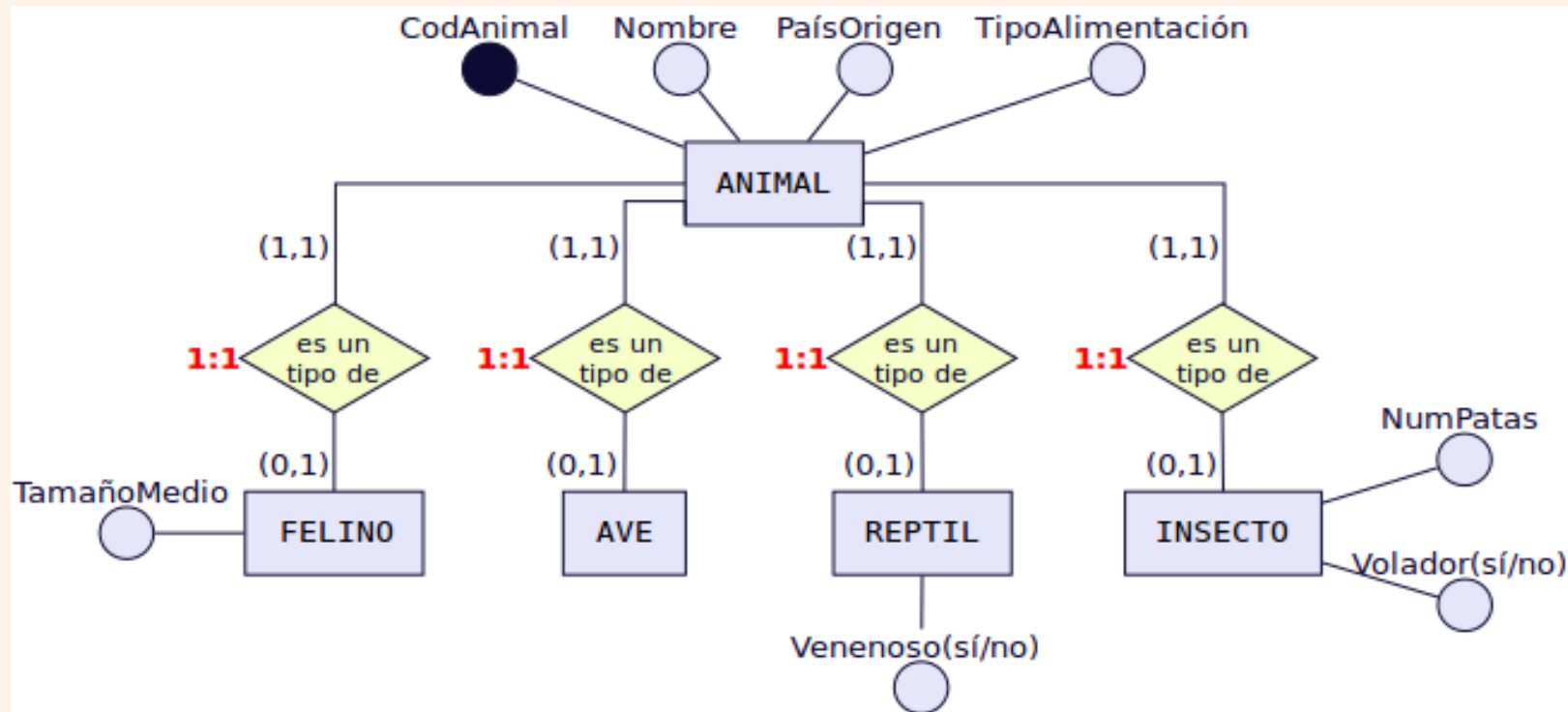
Generalización en el Modelo E/R

Modelo Entidad-Relación



Generalización en el Modelo E/R






Modelo Entidad-Relación





Principios de diseño

Modelo Entidad-Relación

-  Fidelidad
-  Evitar redundancia
-  Lo simple cuenta
-  Elegir la relación adecuada
-  Elegir el tipo de elementos adecuados



Las restricciones del modelado

Modelo Entidad-Relación



Clave



Restricciones de valor único



Restricciones de integridad referencial



Restricciones de dominio

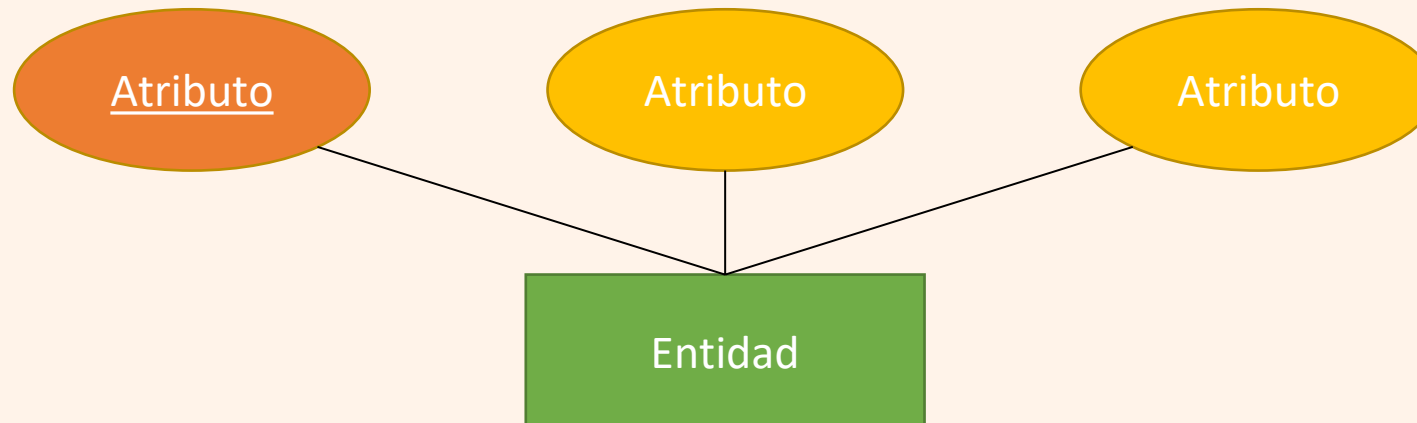


Limitaciones generales

Restricciones - Claves

Modelo Entidad-Relación

Las **claves** son atributos o conjuntos de atributos que identifican de forma única a una entidad dentro de su conjunto de entidades. No hay dos entidades cuyos valores sean iguales para todos los atributos que constituyen una clave. Sin embargo, está permitido que dos entidades coincidan en algunos de los atributos clave, pero no todos.





Restricciones – Valor único

Modelo Entidad-Relación

Las **restricciones de valor único** son requisitos para que el valor en un contexto determinado sea único. Las *claves* son una fuente importante de restricciones de valor único, ya que requieren que cada entidad en un conjunto de entidades tenga valores únicos para los atributos clave. Sin embargo, existen otros motivos de restricciones de valor único, como las relaciones muchos-uno.



Restricciones – Integridad Referencial, de Dominio y Generales

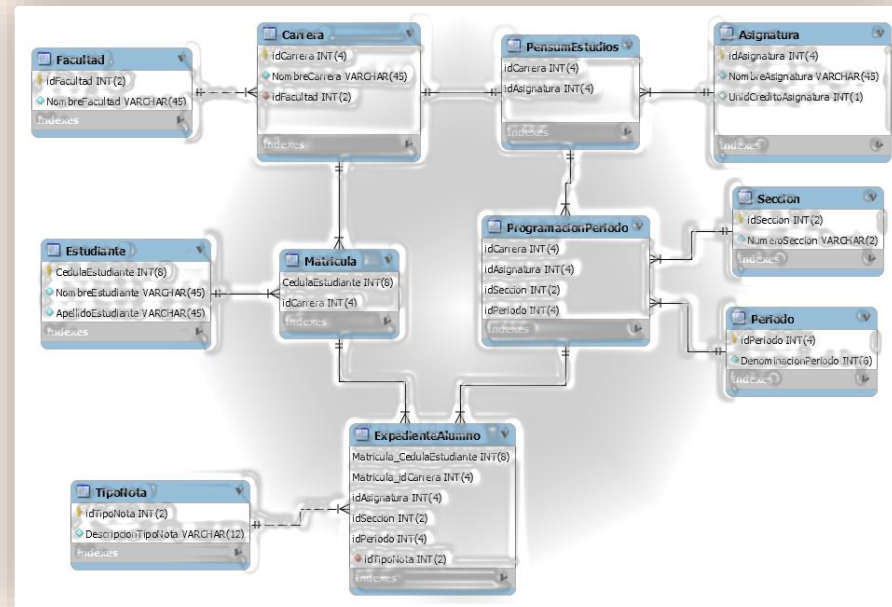
Modelo Entidad-Relación

Las **restricciones de integridad referencial** son requisitos de que un valor al que hace referencia algún objeto exista realmente en la base de datos.

Las **restricciones de dominio** requieren que el valor de un atributo se extraiga de un conjunto específico de valores o se encuentre dentro de un rango específico.

Las **restricciones generales** son afirmaciones arbitrarias que deben mantenerse en la base de datos.

El Modelo de Datos Relacional



Introducción

El Modelo de Datos Relacional

Las implementaciones de bases de datos actualmente son casi todas basadas en el enfoque “Modelo Relacional”. Este es extremadamente útil porque tiene un único concepto de modelado de datos: la “relación”, una tabla de 2 dimensiones donde se encuentran los datos.

Patient				
Code	First Name	Last Name	Birthday	...
HC123	Andrew	Cooper	1992/01/21	...
HC234	Peter	Donely	1975/10/12	...
HC456	Joseph	Orson	1966/06/18	...
HC567	Ann	Watson	1988/09/29	...



Componentes

El Modelo de Datos Relacional

- El modelo relacional nos da una forma única de representar los datos: una tabla de 2 dimensiones llamada “relación”.
- Los **atributos** de una relación sirven de nombres a las columnas de la relación. Generalmente un atributo describe el significado de los datos en la columna.
- El nombre de una relación y el grupo de atributos se llama **esquema** para dicha relación.
- Las filas de una relación (distintas del encabezado que contiene los nombres de los atributos) se llaman **tuplas**. Una tupla tiene un componente por cada atributo de la relación.
- Cada atributo de la relación tiene un **dominio**, es decir, un *tipo* de dato particular.



De E/R a Diseño Relacional

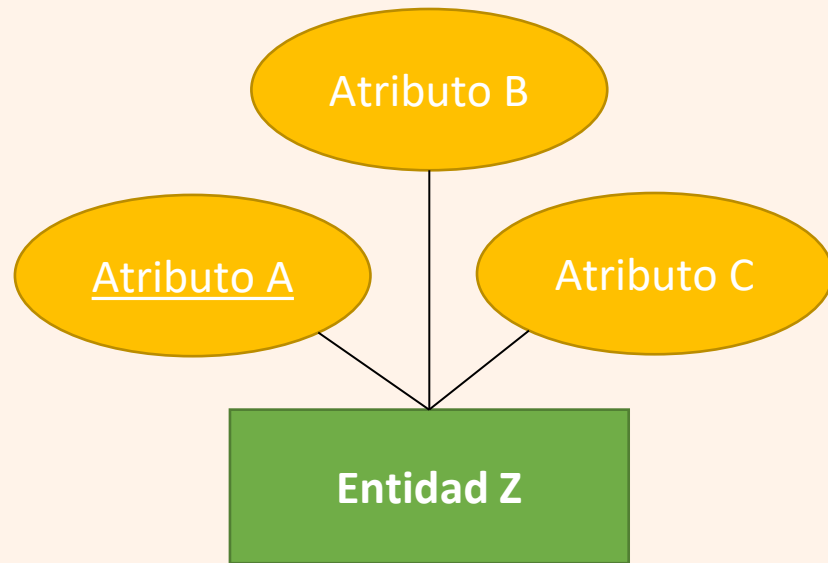
El Modelo de Datos Relacional

- Convertir un diseño E/R en un esquema de base de datos relacional es sencillo:
 1. Convertir cada conjunto de entidades en una relación con el mismo grupo de atributos.
 2. Reemplazar una Relación del diseño en una relación de modelo cuyos atributos son las claves para las entidades conectadas.
- Si bien esto cubre la mayor parte de los casos, hay situaciones especiales que se deben tener en cuenta: entidades débiles, relaciones "isa" (generalización), relaciones uno a muchos

De Entidades a Relaciones

El Modelo de Datos Relacional

- Por cada conjunto de entidades, deberíamos crear una relación con el mismo nombre y grupo de atributos.

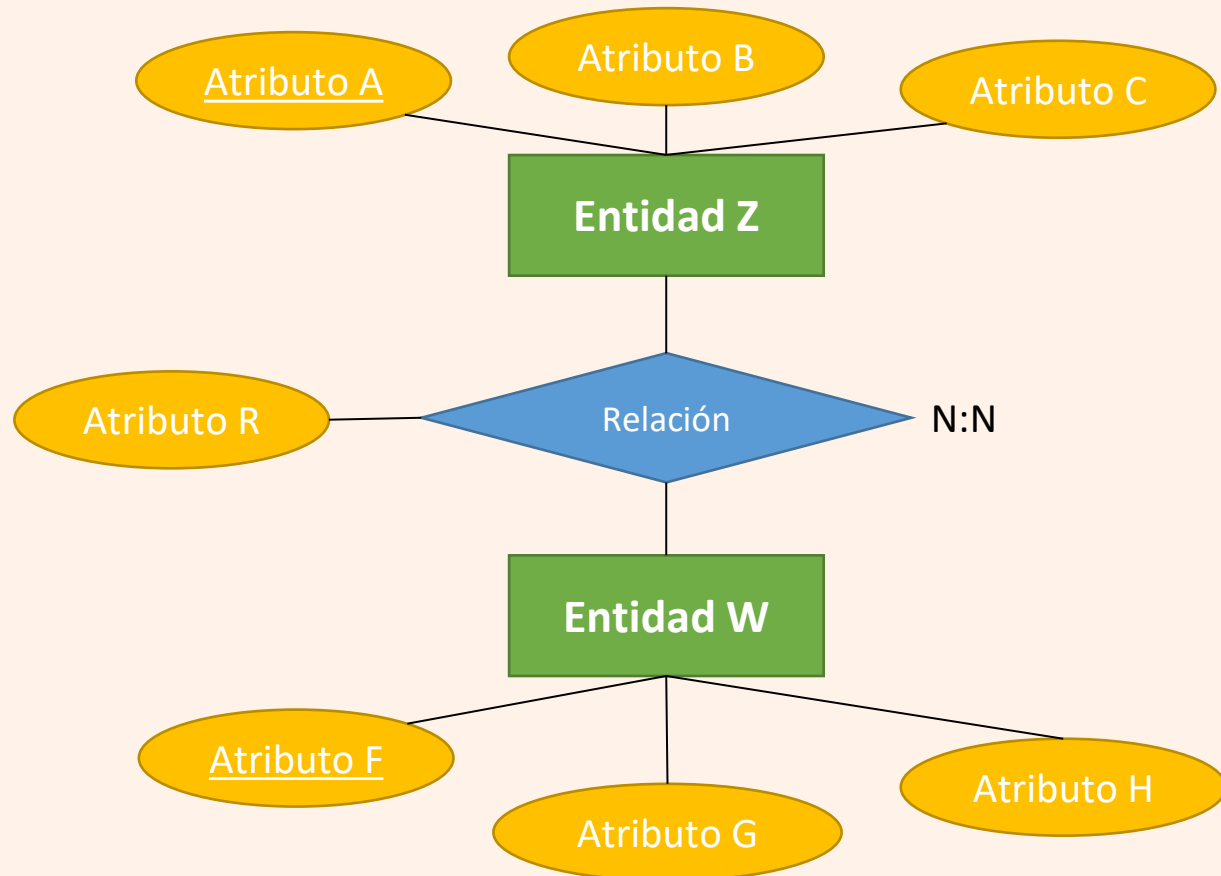


Entidad Z		
<u>Atributo A</u>	Atributo B	Atributo C

De Relaciones E/R a Relaciones

Modelo Entidad/Relación

Las Relaciones del modelo E/R también son representadas por relaciones.



Entidad Z

<u>Atributo A</u>	Atributo B	Atributo C

Relación

<u>Atributo A</u>	<u>Atributo F</u>	Atributo R

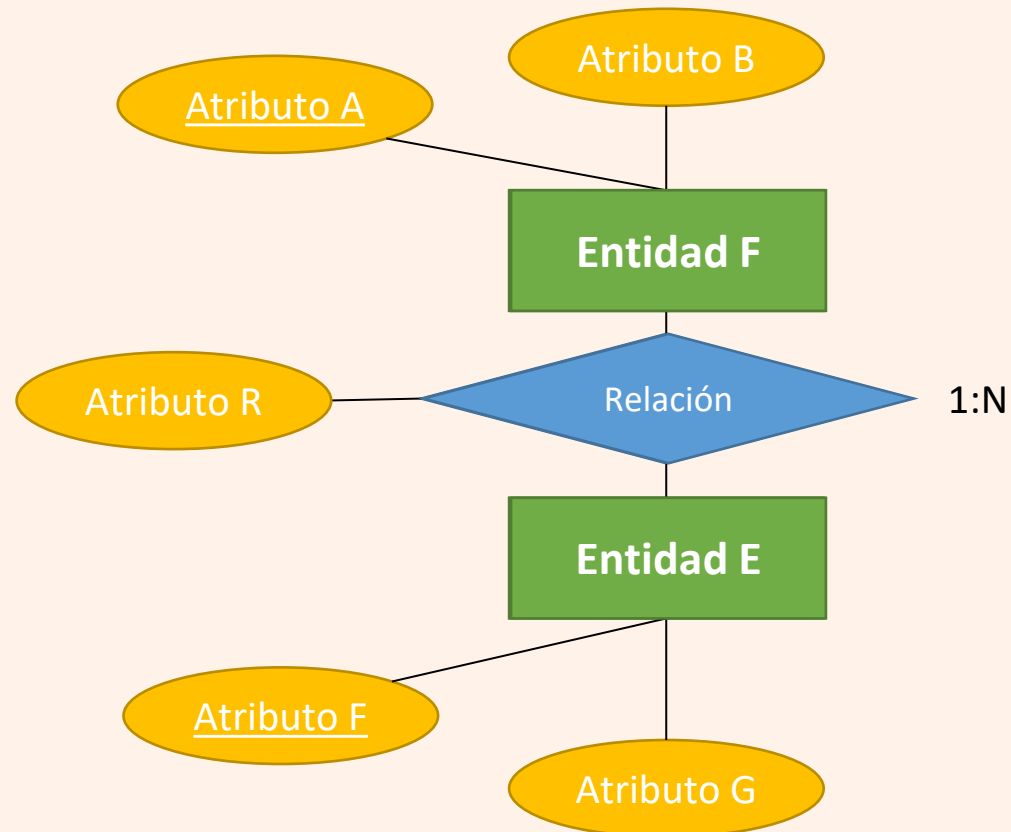
Entidad W

<u>Atributo F</u>	Atributo G	Atributo H

Combinando Relaciones

Modelo Entidad/Relación

Si la cardinalidad es 1:N, la Relación del modelo E/R es combinada con la Entidad de cardinalidad N.



Entidad F	
<u>Atributo A</u>	Atributo B

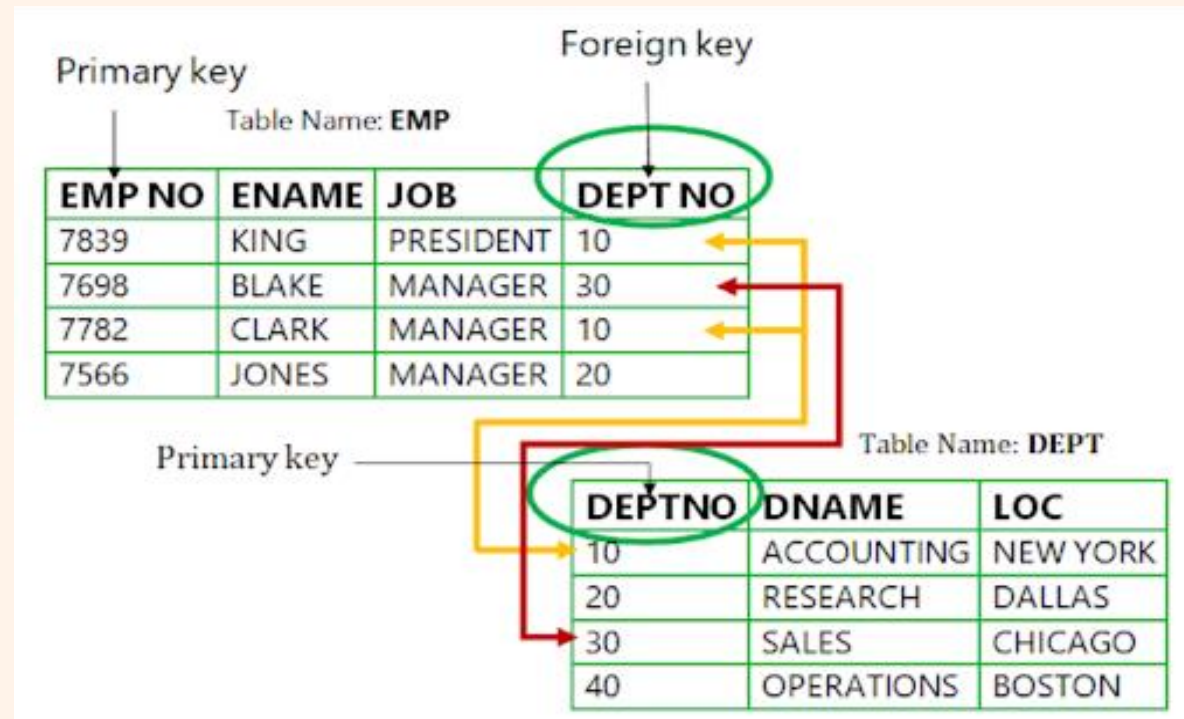
Entidad E			
<u>Atributo F</u>	Atributo G	Atributo R	Atributo A

Ejemplos de Cardinalidad 1:N

Modelo Entidad/Relación

Relación 1 a N entre 2 tablas:

- Cada empleado pertenece a 1 departamento (EMP incluye Foreign key DEPTNO)
- Cada departamento tiene N empleados (DEPT no hace referencia a EMP)

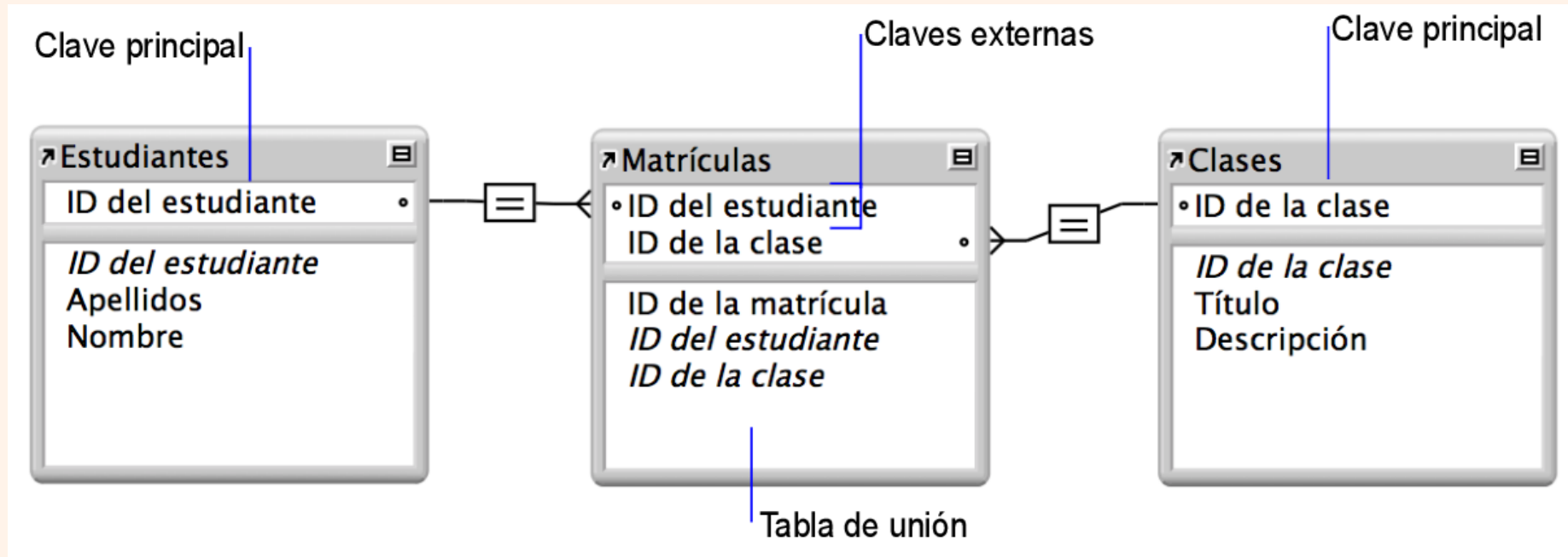


Ejemplos de Cardinalidad N:N

Modelo Entidad/Relación

Relación N a N entre 2 tablas:

- Cada estudiante cursa N materias
- Cada materia es cursada por N estudiantes
- Surge una tabla intermedia con clave compuesta (FK de las tablas principales)





Ingeniería Inversa

Modelo Entidad/Relación

Interpretación de una Base de Datos a partir de su estructura física

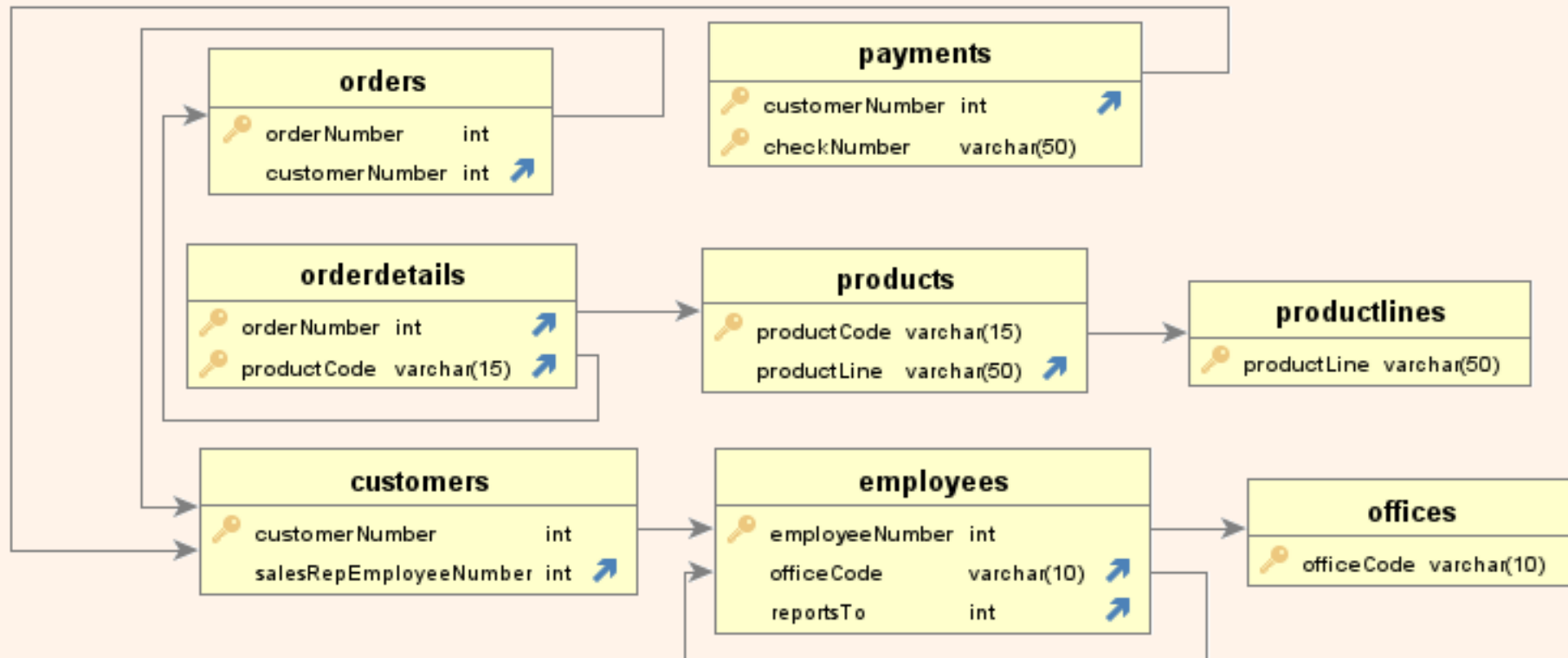
1. Distinguir Tablas “entidades” de Tablas “relaciones N:N”
2. Encontrar e interpretar claves primarias y secundarias
3. Reconocer cardinalidad
4. Interpretar atributos a partir de su nombre y dominio

Análisis de Datos

1. Chequear interpretaciones, realizar cálculos, sacar estadísticas...
2. Crear vistas del tipo “una fila por caso”

Ingeniería Inversa

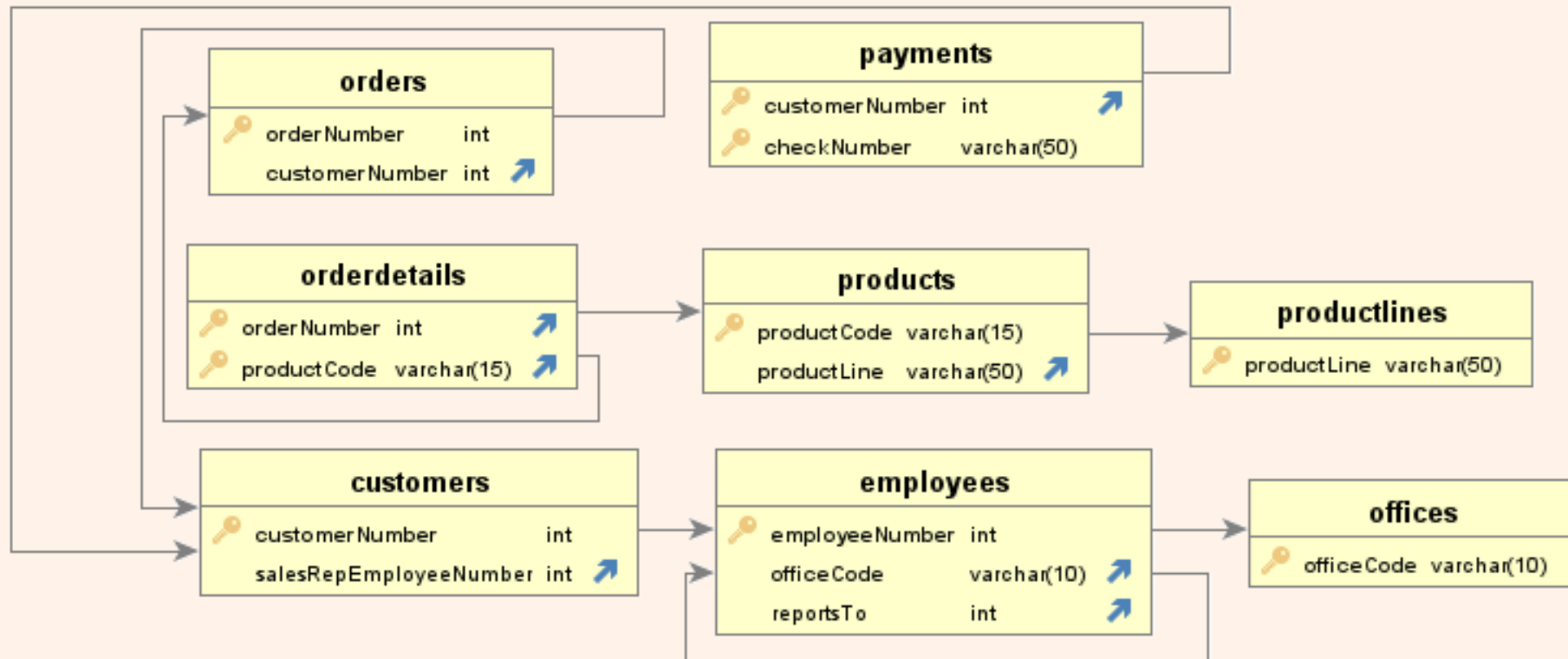
Ejemplo 1



DB: classicmodels
By: DbVisualizer

Ingeniería Inversa

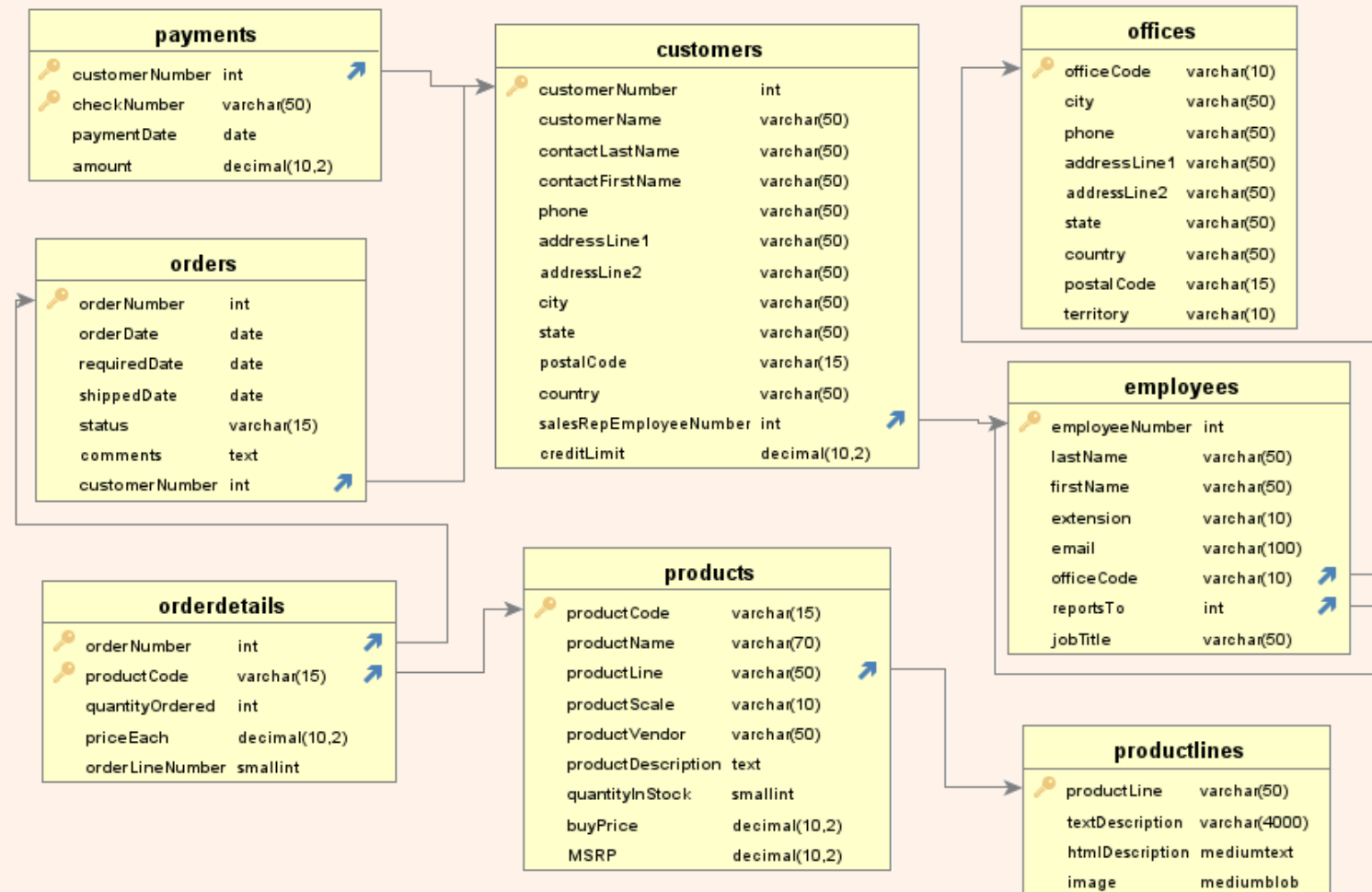
Ejemplo 1 – sólo claves



DB: classicmodels
By: DbVisualizer

Ingeniería Inversa

Ejemplo 1 – detallado




DB: classicmodels
By: DbVisualizer

Ingeniería Inversa

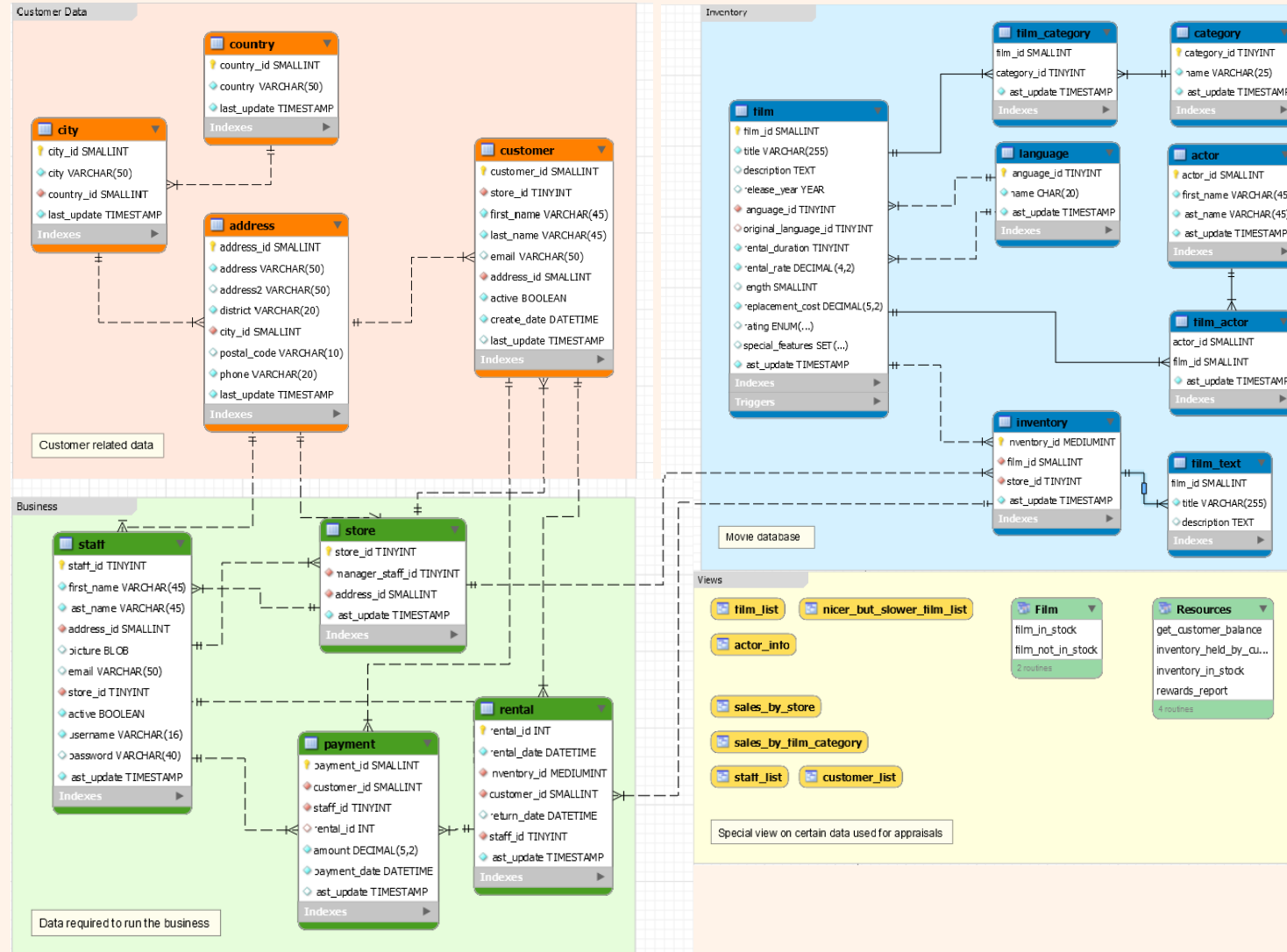
Ejemplo 1 – Detalle tabla

```
CREATE TABLE `customers` (  
  `customerNumber` int(11) NOT NULL,  
  `customerName` varchar(50) NOT NULL,  
  `contactLastName` varchar(50) NOT NULL,  
  `contactFirstName` varchar(50) NOT NULL,  
  `phone` varchar(50) NOT NULL,  
  `addressLine1` varchar(50) NOT NULL,  
  `addressLine2` varchar(50) DEFAULT NULL,  
  `city` varchar(50) NOT NULL,  
  `state` varchar(50) DEFAULT NULL,  
  `postalCode` varchar(15) DEFAULT NULL,  
  `country` varchar(50) NOT NULL,  
  `salesRepEmployeeNumber` int(11) DEFAULT NULL,  
  `creditLimit` decimal(10,2) DEFAULT NULL,  
  PRIMARY KEY (`customerNumber`),  
  KEY `salesRepEmployeeNumber` (`salesRepEmployeeNumber`),  
  CONSTRAINT `customers_ibfk_1` FOREIGN KEY  
    (`salesRepEmployeeNumber`) REFERENCES `employees`  
    (`employeeNumber`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

customers		
	customerNumber	int
	customerName	varchar(50)
	contactLastName	varchar(50)
	contactFirstName	varchar(50)
	phone	varchar(50)
	addressLine1	varchar(50)
	addressLine2	varchar(50)
	city	varchar(50)
	state	varchar(50)
	postalCode	varchar(15)
	country	varchar(50)
	salesRepEmployeeNumber	int
	creditLimit	decimal(10,2)

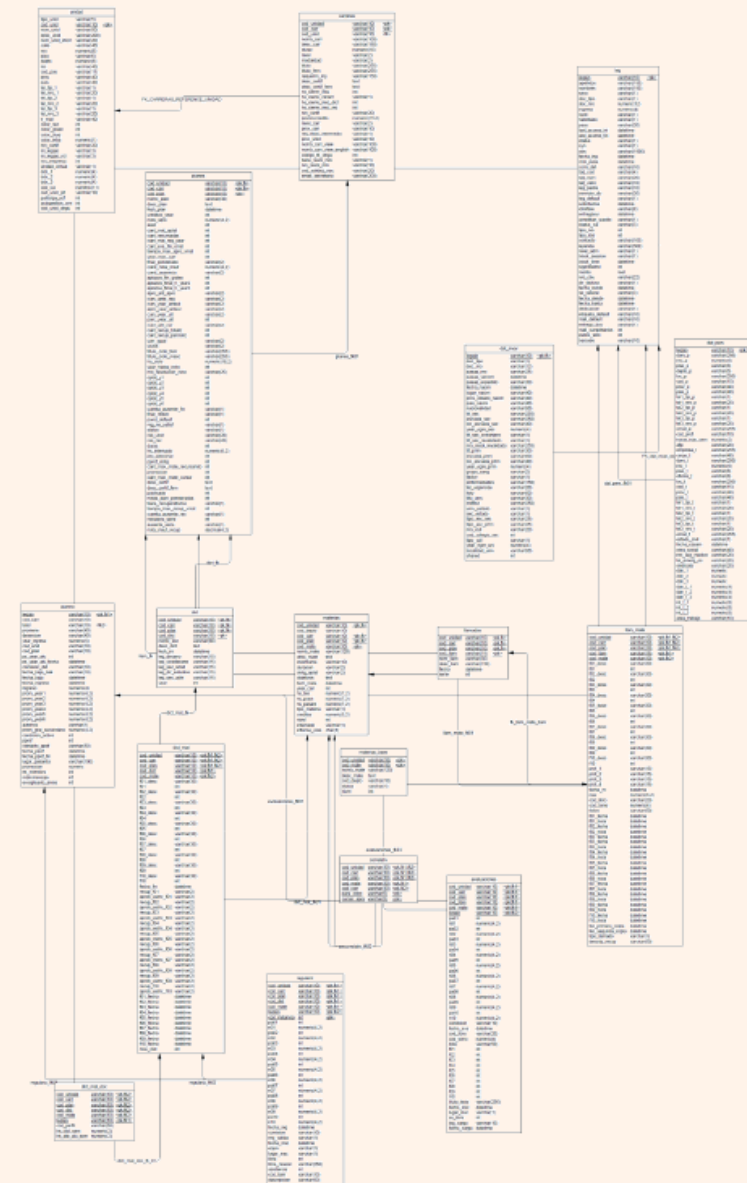
Ingeniería Inversa

Ejemplo 2 – modelo nivel medio



Ingeniería Inversa

Ejemplo 3 – modelo nivel maestría



SOMOSAUSTRAL

Muchas gracias.

www.austral.edu.ar

