

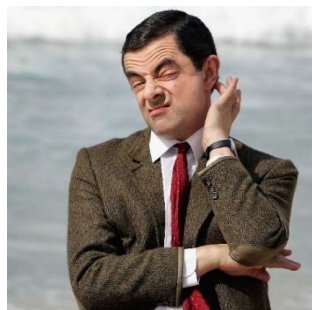


UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE CIENCIAS
FUNDAMENTOS DE BASES DE DATOS

Modelo Relacional

Gerardo Avilés Rosas
gar@ciencias.unam.mx

Modelo relacio... ¿qué?



Años	Sucesos
1970	Surge el Modelo Relacional (E. F. Codd).
1972	Desarrollo teóricos sobre álgebra relacional (Codd).
1973	SEQUEL (D. D. Chamberlin y R. F. Boyce, IBM) .
1974	Ingres (Universidad de Berkeley).
1977	Sistema R (Laboratorios San José, IBM)
1979	Oracle (Relational Software, Inc, implementa SEQUEL)
1986	ANSI SQL (primera versión estándar, SQL-86)
1990	Modelo Relacional versión 2 (RM/V2) Codd
1992	Revisión mayor (SQL2)
1999	SQL2000 (características de OO)

Information Retrieval

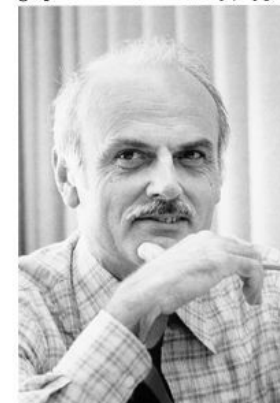
A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

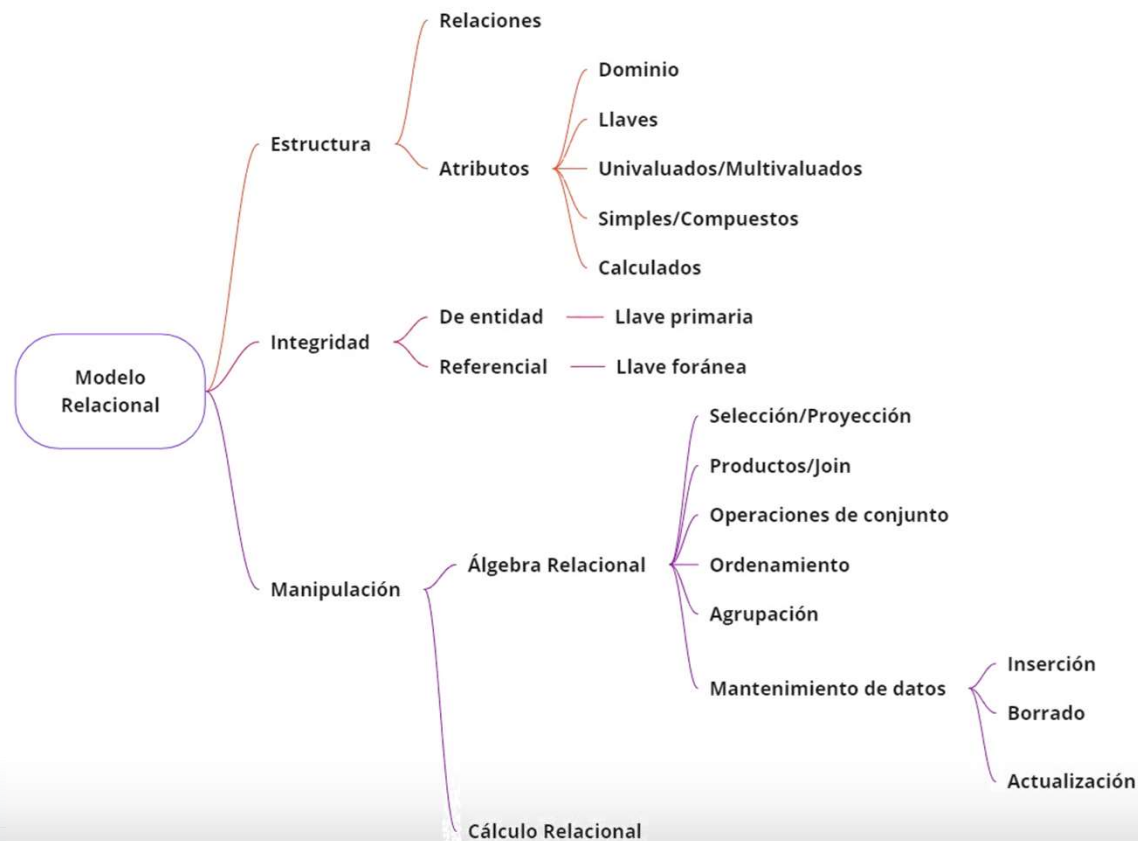
Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network

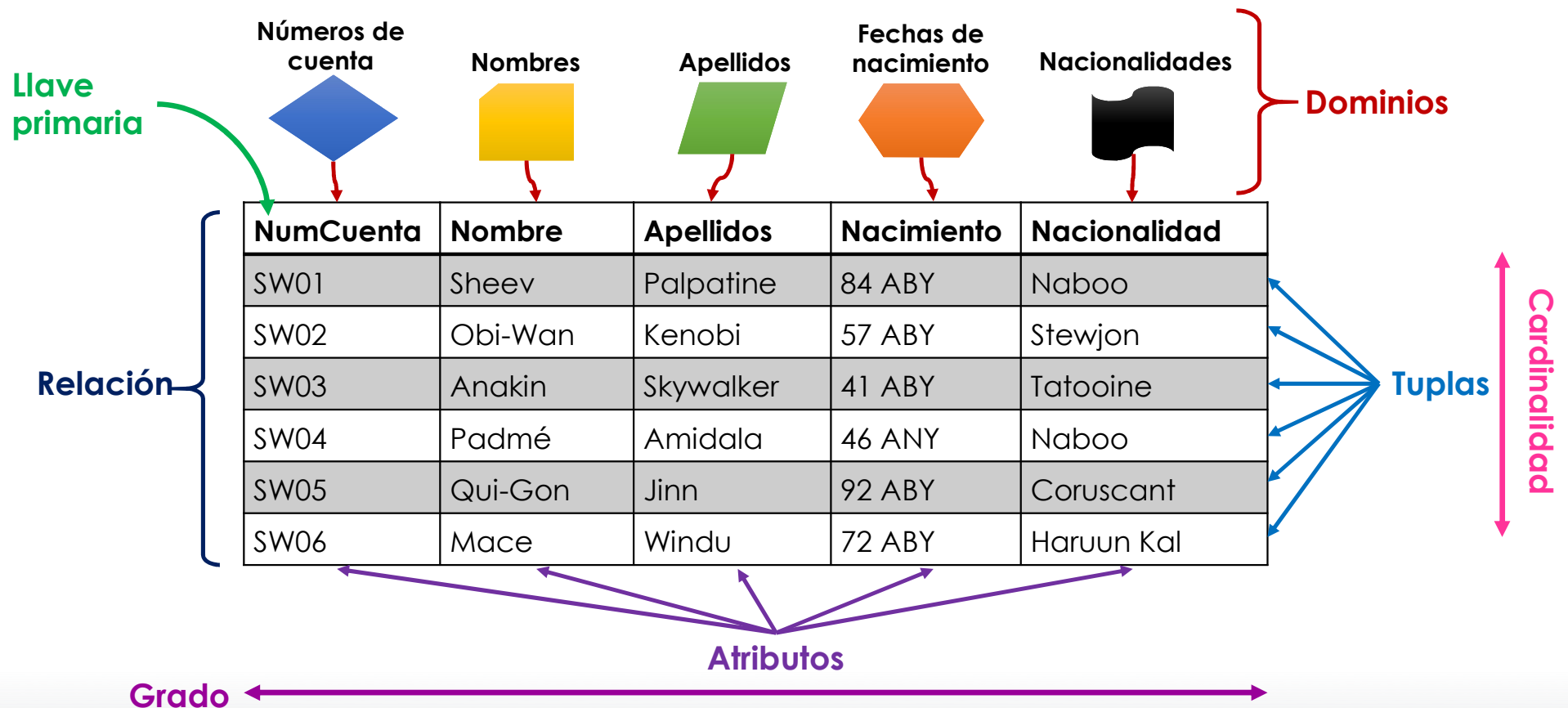
The relational view (or model Section 1 appears to be superior in graph or network model [3, 4] pre



Está basado en dos ramas de las matemáticas: la **teoría de conjuntos** y la **lógica de predicados** de primer orden. Su base matemática hace que el modelo sea **predecible, fiable y seguro** ¿qué más se puede pedir?



Una BDR (**Codd, 1968**) es una BD formada por una colección de relaciones:



- **Dominio:** Conjunto finito de valores **homogéneos** y **atómicos** caracterizados por un nombre.
- **Atributo:** Aquel que participa en la descripción de las entidades y que como tal constituye una pieza específica de información para un determinado dominio.
- **Llaves:** Conjunto no vacío de atributos que identifican unívoca y mínimamente cada *tupla*:
 - ❑ **Llave primaria:** Aquella llave que permite identificar *tuplas* de la relación de forma única.
 - ❑ **Llaves candidatas:** Son aquellas que no han sido escogidas como llaves primarias pero que también podrían identificar de manera única a una *tupla*.
 - ❑ **Llave foránea:** Conjunto no vacío de atributos cuyos valores han de coincidir con los valores de la llave primaria en una relación.
- **Restricciones:** Son estructuras no permitidas y hay de dos tipos: *inherentes* y *del usuario*.

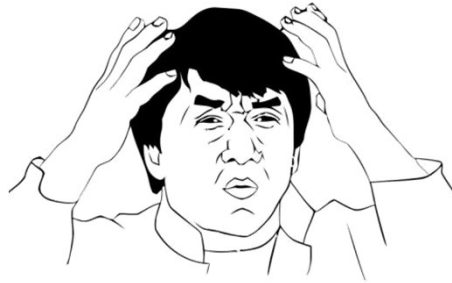
- Por ejemplo, la relación **alumno(nombre,num_cta,edad,...)** o bien **alumno(num_cta,...,nombre,...,edad,...)**



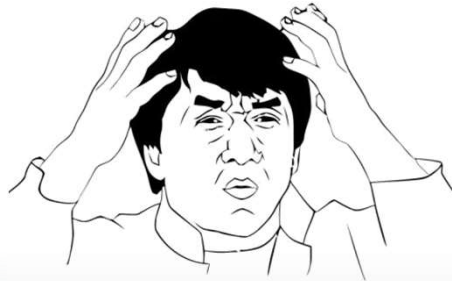
Objetivos del modelo relacional

- **Independencia física.** El modo en el que se almacenan los datos no influye en su manipulación lógica y por tanto, los usuarios que acceden a esos datos no tienen que modificar sus programas por cambios en el almacenamiento físico.
- **Independencia lógica.** El añadir, eliminar o modificar objetos de la base de datos no repercute en los programas y/o usuarios que están accediendo a subconjuntos parciales de los mismos (vistas).
- **Flexibilidad.** En el sentido de poder presentar a cada usuario los datos de la forma en que éste prefiera.
- **Uniformidad.** Las estructuras lógicas de los datos presentan un aspecto uniforme, lo que facilita la concepción y manipulación de la base de datos por parte de los usuarios.
- **Sencillez.** Las características anteriores, así como unos lenguajes de usuario muy sencillos hace que este modelo sea fácil de comprender y de utilizar por parte del usuario final.

Una relación R de los conjuntos $A_1, A_2, A_3, \dots, A_n$ es un subconjunto del **producto cartesiano** de los mismos: $R \subseteq A_1 \times A_2 \times A_3 \times A_n$



- **Producto cartesiano:** El producto cartesiano de los conjuntos $A_1, A_2, A_3, \dots, A_n$ representado por $A_1 \times A_2 \times A_3 \times A_n$ es el conjunto de todas las tuplas ordenadas $(a_1, a_2, a_3, \dots, a_n)$ tales que $a_1 \in A_1, a_2 \in A_2, a_3 \in A_3, \dots, a_n \in A_n$.



Vamos a suponer que tenemos información de **ropa (A_1)**, **colores (A_2)** y **tonos (A_3)**:

- ❑ $A_1 = \{\text{camisa, zapatos, calcetines, pantalón}\}$
- ❑ $A_2 = \{\text{rojo, azul, verde, amarillo}\}$
- ❑ $A_3 = \{\text{claro, intenso, oscuro, brillante}\}$

Es posible definir las siguientes relaciones:

- $R_1 = \{(\text{camisa, azul, claro}), (\text{zapatos, verde, brillante}), (\text{calcetines, rojo, intenso}), (\text{pantalón, azul, oscuro})\}$
- $R_2 = \{(\text{camisa, rojo, claro}), (\text{camisa, rojo, intenso}), (\text{zapatos, rojo, claro}), (\text{zapatos, rojo, oscuro})\}$
- $R_3 = \{(\text{pantalón, verde, claro}), (\text{camisa, verde, oscuro})\}$



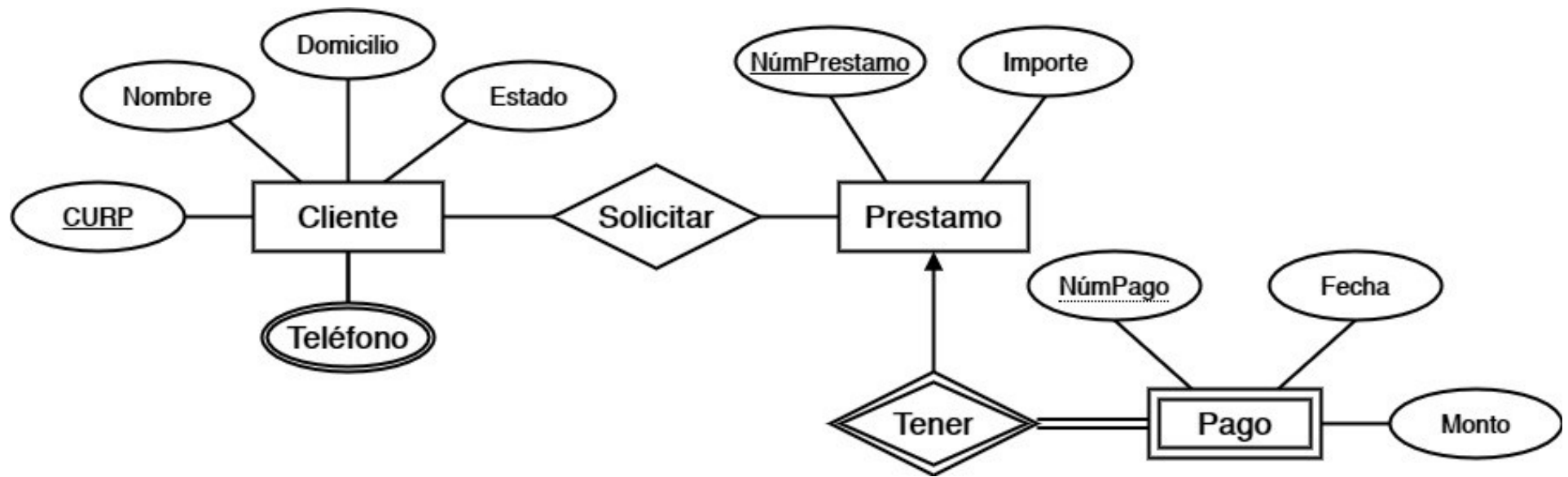
Conversión del modelo E-R a Modelo Relacional

- Las llaves permiten que los conjuntos de entidades y los de relaciones se expresen de manera uniforme como **tablas** que representan el contenido de la BD.
- Una BD que conforma un diagrama E-R puede representarse como una **colección de tablas**.
- Para cada conjunto de entidades y cada conjunto de relaciones existe una **tabla única** con el nombre de tal conjunto.
- Cada tabla tiene una un **número columnas fijo**, cada una con nombre único.
- Convertir un diagrama E-R a un formato de tablas es la base para derivar un diseño de base de datos relacional a partir de un diagrama E-R



NumCuenta	Nombre	Apellidos	Nacimiento	Nacionalidad
SW01	Sheev	Palpatine	84 ABY	Naboo
SW02	Obi-Wan	Kenobi	57 ABY	Stewjon
SW03	Anakin	Skywalker	41 ABY	Tatooine
SW04	Padmé	Amidala	46 ANY	Naboo
SW05	Qui-Gon	Jinn	92 ABY	Coruscant
SW06	Mace	Windu	72 ABY	Haruun Kal

Ejemplo de transformación





Entidades fuertes

Una **entidad fuerte** se convierte en una tabla con los mismos atributos.

Cliente:

CURP	Nombre	Domicilio	Estado
LOOC770826MMNPRR12	Carmen López Ortega	Azucena 456	Michoacán
RIOG770826HCXSRB15	Gabriel Ortiz Rios	Av. Montejo 124	CDMX
MEOJ770826HVZNRV43	Javier Ortega Mendoza	Jazmín No. 58	Veracruz
JUCG770826MGRRRD58	Guadalupe Cruz Juárez	Arenal Núm. 234	Guerrero
PEEC770826HMSRSR23	Carlos Espinoza Pérez	Goya No. 896	Morelos
MOVJ770826HSRRGS97	Jesús Vega Morales	Sán Andrés 547	Sonora
MUER770826HJCNSS89	Rosa María Espinoza Muñoz	Av. San José 89	Jalisco
GURS770826MTCTSL63	Salvador Ríos Gutiérrez	Zafiro 587	Tabasco
VAGS770826MNTLMR16	Sergio Gómez Valdez	Av. Pacífico 98	Nayarit
JIPA770826HPLMNN98	Andrés Peña Jiménez	Del Carmen 78	Puebla

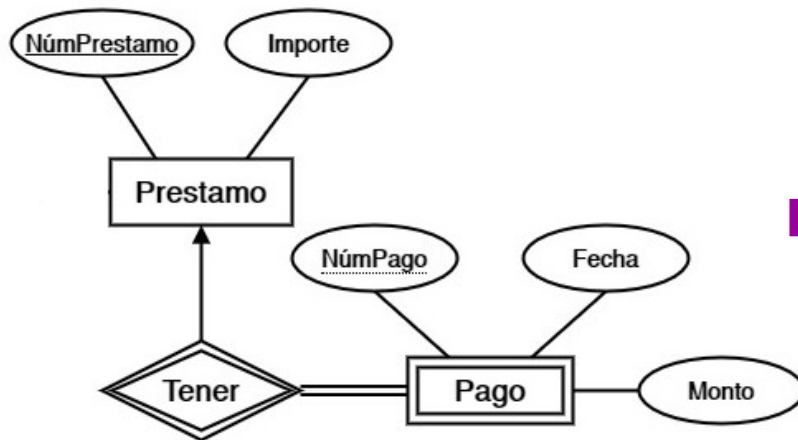
Préstamo:

NúmPréstamo	Importe
P-17	9,253
P-23	16,736
P-25	13,494
P-14	20,548
P-93	14,858
P-11	17,761
P-22	30,395
P-15	31,103
P-67	18,481
P-24	33,922

Nota: Si la entidad tiene **atributos multivaluados**, estos no se consideran en la relación resultante.

Sea **A** una **entidad débil** con atributos a_1, a_2, \dots, a_n y **B** la **entidad fuerte** (de la que depende A), con **llave primaria** formada por los atributos b_1, b_2, \dots, b_m . Esta entidad débil se representa mediante una **tabla C** con columnas para cada atributo del conjunto

$$\{a_1, a_2, \dots, a_n\} \cup \{b_1, b_2, \dots, b_m\}$$



Pago:

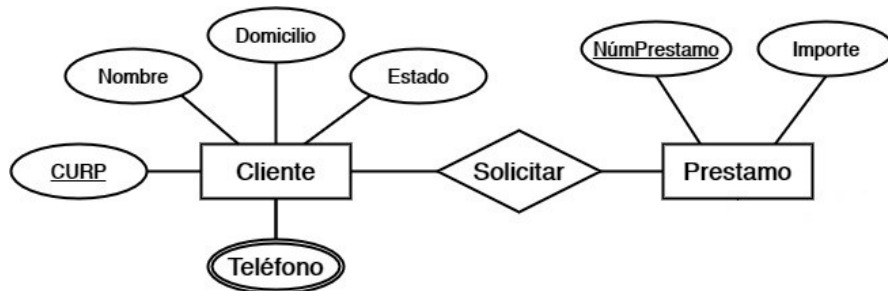
NúmPrestamo	NumPago	Fecha	Monto
P-17	5	25/07/2013	568
P-23	11	15/02/2014	1272
P-25	22	07/03/2014	550
P-14	14	01/04/2014	1431
P-93	5	16/04/2014	740
P-11	13	18/05/2014	1904
P-22	14	16/07/2014	1672
P-15	10	12/12/2014	1280
P-67	45	29/01/2015	1439
P-24	22	16/10/2015	1811

Nota: La llave primaria de la tabla C es una llave compuesta por el **discriminante** (NúmPago) y la **llave primaria** de su **entidad propietaria** (NúmPréstamo).

Sean:

- **R** un tipo de relaciones,
- $\{a_1, a_2, \dots, a_n\}$ el conjunto de atributos formados por la **unión de llaves primarias** de cada uno de los conjuntos de entidades que participan en **R**,
- $\{r_1, r_2, \dots, r_m\}$ el conjunto de atributos de **R** (si existen).

El **tipo de relación** se representa mediante una tabla llamada **R** con una columna por cada atributo de $\{a_1, a_2, \dots, a_n\} \cup \{r_1, r_2, \dots, r_m\}$

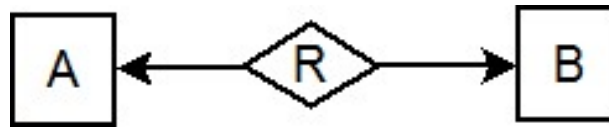


Solicitar:

CURP	NúmPrestamo
LOOC770826MMNPRR12	P-17
RIOG770826HCXSRB15	P-23
MEOJ770826HVZNRV43	P-25
JUCG770826MGRRRD58	P-14
PEEC770826HMSRSR23	P-93
MOVJ770826HSRRGS97	P-11
MUER770826HJCNSS89	P-22
GURS770826MTCTSL63	P-15
VAGS770826MNTLMR16	P-67
JIPA770826HPLMNN98	P-24

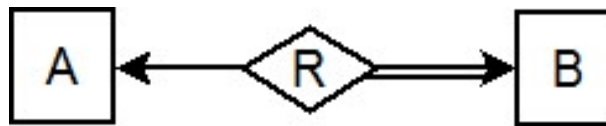
Para **relaciones 1:1** se tienen tres posibilidades:

- **Relación parcial**



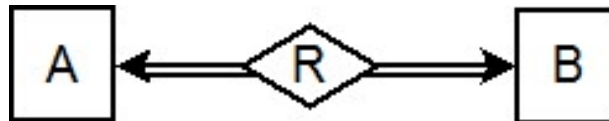
Se sigue la misma regla que para relaciones con **cardinalidad N:M**, pero se pierde semántica.

- **Relación total de un lado**



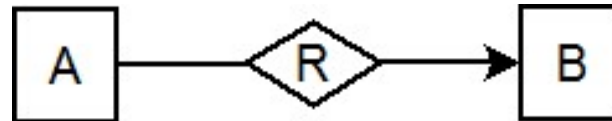
Incluir en **B** los atributos de **R** y la llave de **A**.

- **Relación total de ambos lados**



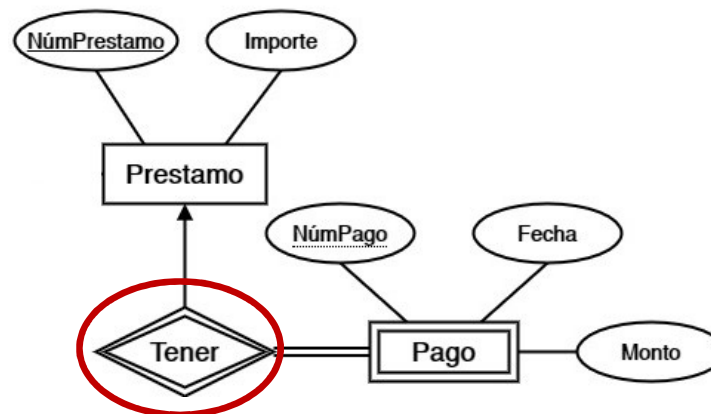
Incluir los atributos de **A**, **B** y de **R** en una sola relación adicional.

- Para **relaciones 1:N** como la siguiente:



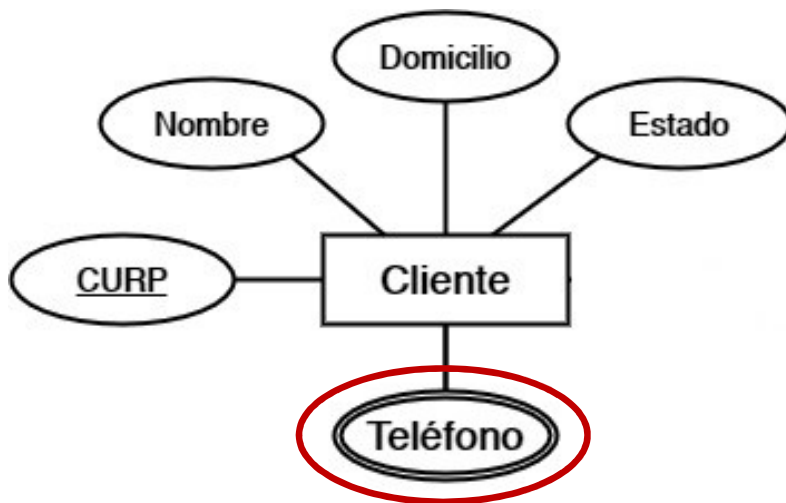
En la relación **A** se incluye la llave de la relación **B** más los atributos de la relación **R**.

- Un **tipo de relación** que asocia un conjunto de **entidades débiles** a un conjunto de **entidades fuertes**:



En general, la tabla para esta relación **es redundante** y por tanto no necesita representarse de esta manera.

- Los **atributos multivaluados** se convierten en **tablas** no en columnas.
- Si **M** es un **atributo multivaluado**, se crea una tabla **T** con una columna que corresponde a la llave primaria del **tipo de entidad** o **tipo de relación** del que **M** es atributo y otra para el atributo.
- Se establecen ambos atributos como llave primaria de la tabla.

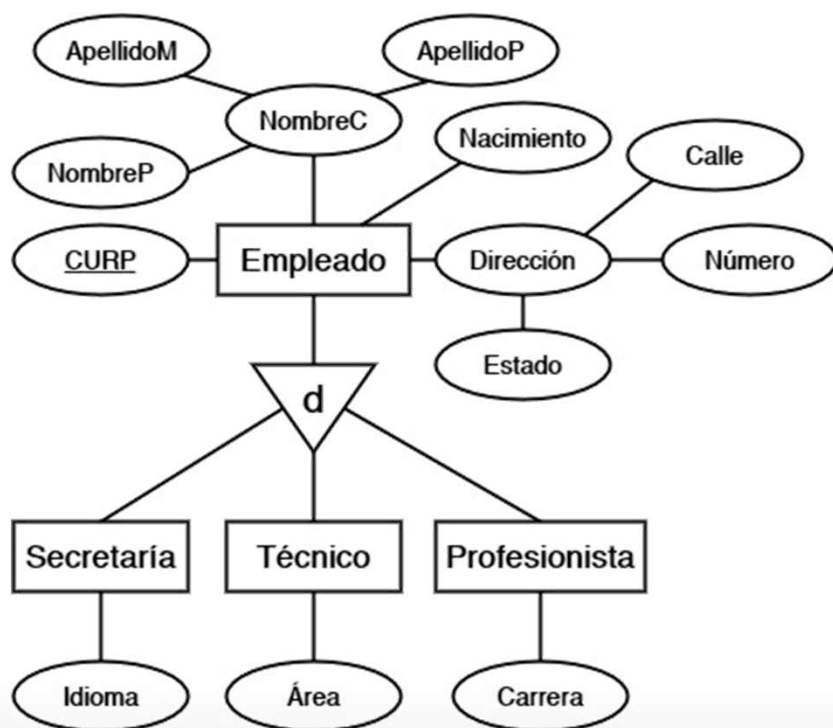


Teléfono

CURP	Teléfono
LOOC770826MMNPRR12	5545876987
RIOG770826HCXSRB15	5514782356
MEOJ770826HVZNRV43	5587982514
MEOJ770826HVZNRV43	5536214578
PEEC770826HMSRSR23	5523654789
PEEC770826HMSRSR23	5541247896
MUER770826HJCNSS89	5565897412
GURS770826MTCTSL63	5562457891
GURS770826MTCTSL63	5547268443
JIPA770826HPLMNN98	5594756214

Generalización/Especialización: Parcial – disyunción

- Se crea una tabla para la **superentidad** con todos sus atributos.
- Se crea una tabla **T_i** para cada **subentidad** que contenga los atributos de esa **subentidad** más el **atributo llave** de la entidad superior.



Empleado

(CURP, NombreP, ApellidoP, ApellidoM, Nacimiento, Calle, Número, Estado)

Secretaría

(CURP, Idioma)

Técnico

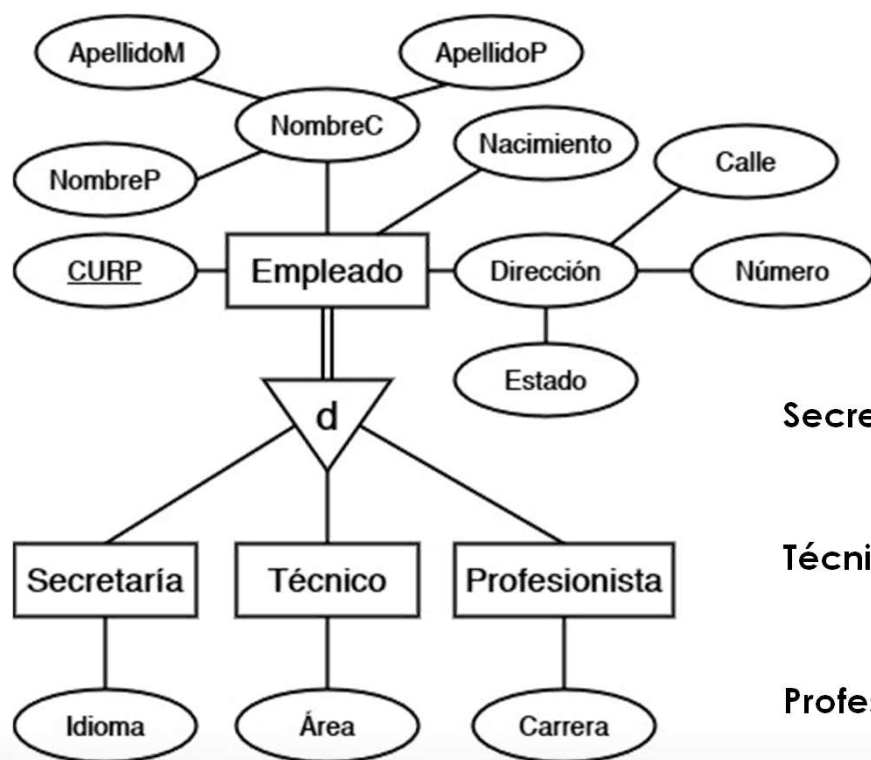
(CURP, Área)

Profesionista

(CURP, Carrera)

Generalización/Especialización: Total – disyunción

Se crea una tabla T_i para cada **subentidad** con sus atributos más los de la **superentidad**. La **llave** es la misma de la superentidad.



Secretaria

(CURP, NombreP, ApellidoP, ApellidoM, Nacimiento, Calle, Número, Estado, Idioma)

Técnico

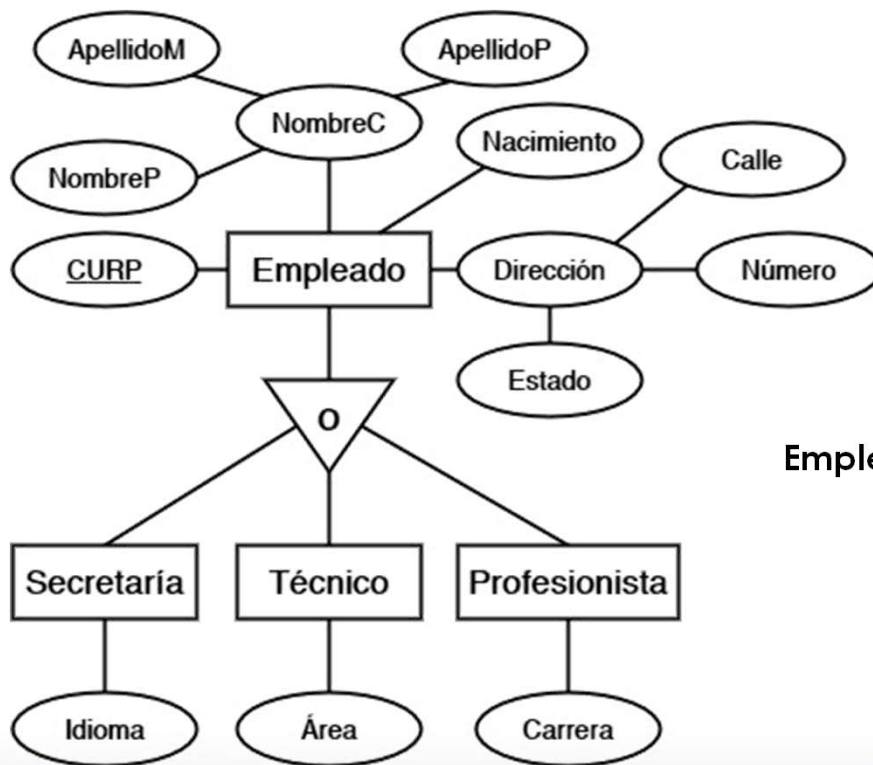
(CURP, NombreP, ApellidoP, ApellidoM, Nacimiento, Calle, Número, Estado, Área)

Profesionista

(CURP, NombreP, ApellidoP, ApellidoM, Nacimiento, Calle, Número, Estado, Carrera)

Generalización/Especialización: Traslape

Se crea una relación T_i con un conjunto de atributos de **tipo booleano** para saber si se tiene ese **tipo de especialización**.

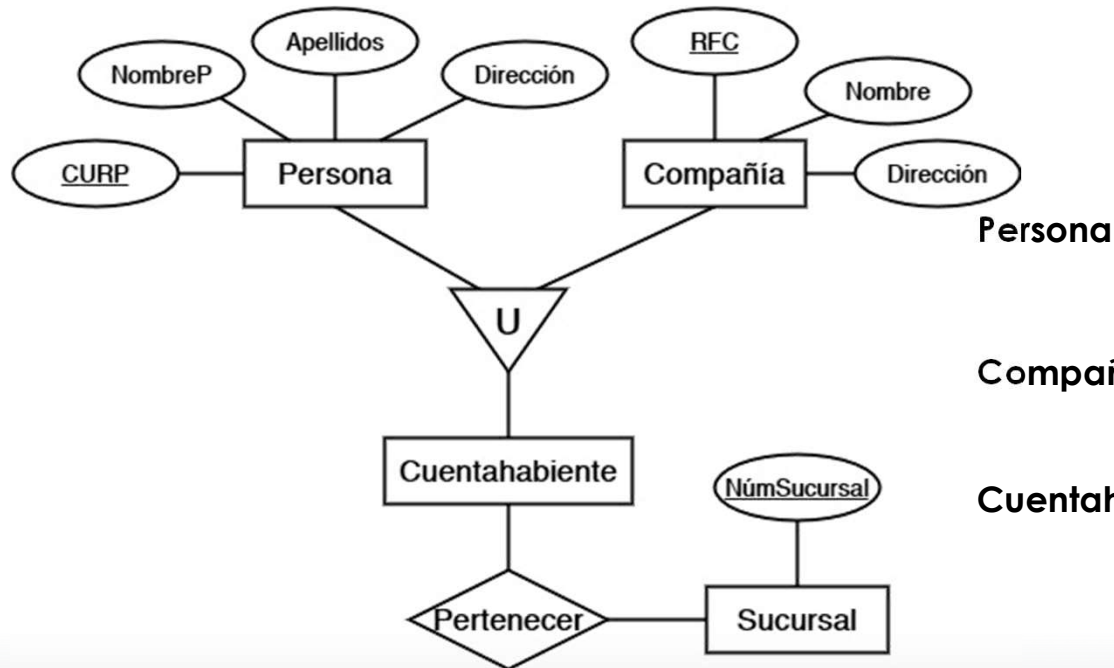


Empleado

(**CURP**, NombreP, ApellidoP, ApellidoM, Nacimiento, Calle, Número, Estado, Idioma, Área, Carrera, esSecretaria, esTécnico, esProfesionista)

Unión: diferentes llaves primarias

- Se crea una **relación R** que corresponda a la categoría y se le asigna una **llave sustituta** arbitraria.
- Se añade la llave sustituta a modo de **llave foránea** en cada una de las relaciones R_i que corresponden a las **superentidades** de la categoría.



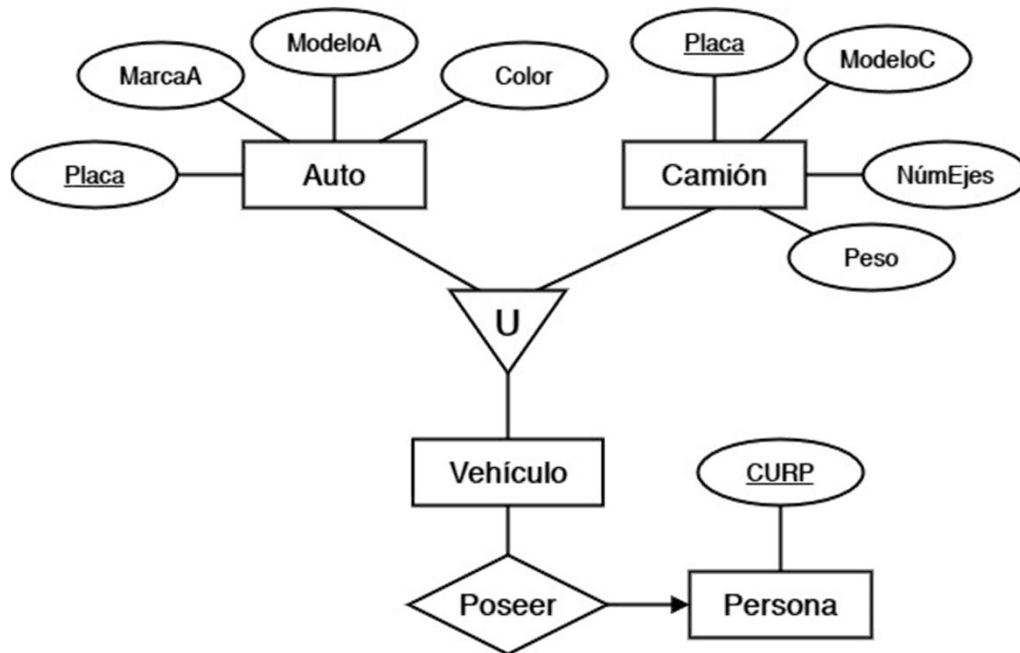
(CURP, NombreP, Apellidos, Dirección, **IdCuentahabiente**)

(RFC, Nombre, Dirección, **IdCuentahabiente**)

Cuentahabiente (IdCuentahabiente)

Unión: mismas llaves primarias

Se crea una **relación R** que corresponda a la **categoría** y se le asigna como atributo **llave primaria** la **llave común** a todas las **superclases** de la categoría.



Auto (Placa, MarcaA, ModeloA, Color)

Camión (Placa, MarcaC, ModeloC, NúmEjes, Peso)

Vehículo (Placa)

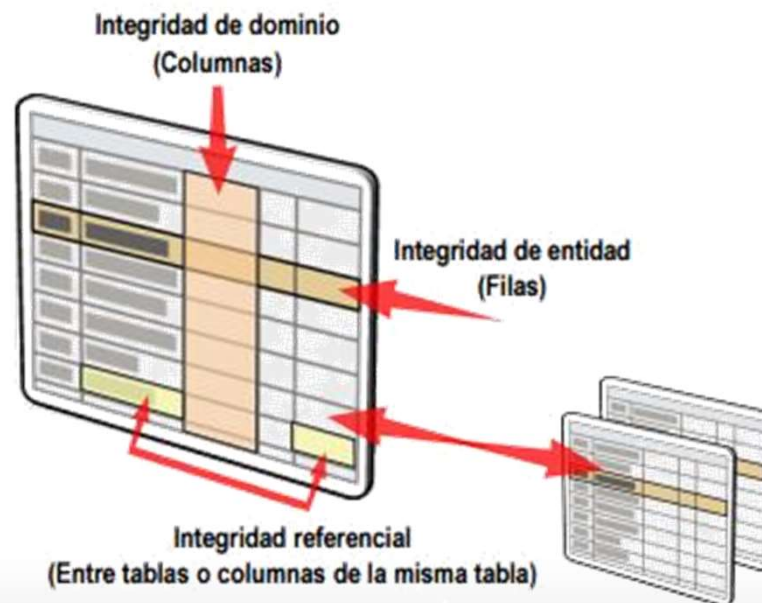


Restricciones de integridad

- **Conjunto de reglas** que **limitan** los valores que son aceptables o acciones, cuyo objetivo es facilitar la **exactitud e integridad** de los datos.
 - Existen **tres tipos** de restricciones:
 - ❑ **Restricciones de dominio**
 - ❖ **Todos los valores** que aparecen en una columna de una relación deben ser del **mismo dominio**.
 - ❖ Un **dominio** es el **conjunto de valores** que **pueden ser asignados** a un atributo.
 - ❑ **Integridad de la entidad**
 - ❖ **Regla diseñada para asegurar** que toda relación tiene una **llave primaria** y que los valores que se tiene para ella son **todos válidos**.
 - ❖ **Garantiza** que los atributos que forman la llave primaria **no sean nulos**.
 - ❖ **Valores nulos**. En ocasiones se debe asignar un valor para un componente de una tupla, pero no se puede decir cuál. En este caso se asigna el valor nulo **NULL**. Existen diferentes interpretaciones para ellos: valor desconocido, valor inaplicable o valor perdido.
- En general, un atributo podría permitírsele o no tener valor nulo.

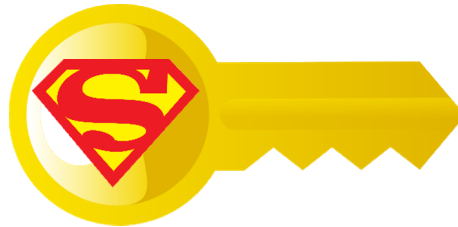
❑ Integridad referencial

- ❖ **Regla** que mantiene la **consistencia entre las tuplas** de dos relaciones.
- ❖ Si existen una **llave externa (FK)** en una relación, **cada valor** de esa llave **debe coincidir** con su correspondiente valor para la **llave primaria (PK)** en otra relación o el valor debe ser nulo.
- ❖ Se establecen **políticas de mantenimiento** para las llaves foráneas.



Restricciones de integridad: Llaves

- Una **llave** es un conjunto **no vacío** de atributos que identifican de manera única a cada tupla.
- Una **superllave** es cualquier conjunto de atributos cuyos valores no se repiten en tuplas distintas de la relación.



<u>StudentId</u>	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042
L0023487	Peter	Murray	P301
L0018453	Anne	Norris	S042

... Restricciones de integridad: Llaves

- Sea **R una relación**, entonces una **llave candidata** para **R** es un subconjunto del conjunto de atributos de **R**, digamos **k** tal que:
 - Unicidad.** No existe par de tuplas distintas en R con el mismo valor para K.
 - Irreductibilidad.** Ningún subconjunto propio de K tiene la propiedad de unicidad.
- Una llave candidata que involucra más de un atributo se llama **compuesta**, en otro caso se llama **simple**.



<u>StudentId</u>	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042
L0023487	Peter	Murray	P301
L0018453	Anne	Norris	S042

CANDIDATE KEY	COMPOSITE KEY
A super key with no redundant attributes	A key that consist of two or more attributes that uniquely identify any row in the table
Candidate key can have one attribute	Composite key must have a minimum of two attributes
In a table with attributes id, name and phone; id and phone are the candidate keys	In a table with attributes student_id, subject_id, marks and exam_name; the composite key is the combination of student_id and subject_id
	Visit www.PEDIAA.com

... Restricciones de integridad: Llaves

Una **llave primaria** es una llave candidata elegida, casi siempre, arbitrariamente.



<u>StudentId</u>	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042
L0023487	Peter	Murray	P301
L0018453	Anne	Norris	S042

... Restricciones de integridad: Llaves

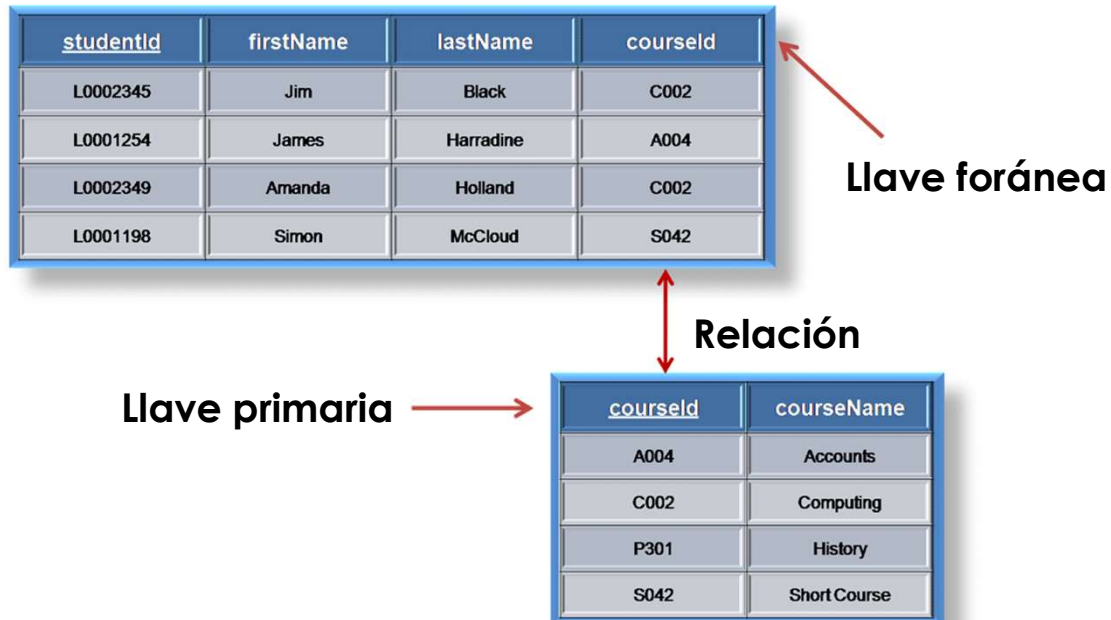
- Cuando una **llave c** en una **relación R** aparece como **atributo** en otra **relación S**, se dice que **c** es una **llave externa (foránea)** en **S**.
- Sea **R₂** una relación, se define como una **llave externa** en **R₂** como un **subconjunto FK**, del conjunto de atributos de **R₂**, tales que:
 - ❑ Existe una relación **R₁** con una **llave PK**, y
 - ❑ Cada valor de **FK** en **R₂** es **idéntico** al valor de **PK** en alguna tupla de **R₁**.
- Terminología:
 - ❑ El valor de una **llave externa** representa una **referencia** a la tupla que contiene el valor de la llave.
 - ❑ El problema de asegurar que la base de datos no incluye cualquier **llave no válida** se conoce como problema de **integridad referencial**.





Restricciones de integridad referencial

- La base de datos **no debe contener** valores de **llave externa** que no se correspondan con un valor de la llave candidata (Si **B** referencia a **A**, entonces **A** debe existir).
- Llave externa** e **integridad referencial**, se definen una en términos de la otra.



- Este es un concepto relacionado con las **entidades débiles**, que como se recordará, tienen una llave primaria compuesta.
- Se crean para **simplificar la estructura** de las llaves a partir de un **número serial** u otra configuración.
- **Hoberman (2006)** recomienda crear una llave sustituta cuando:
 - ❑ Se tiene una **llave primaria compuesta**.
 - ❑ Cuando se tiene una **llave primaria natural** (llave utilizada en la organización) y esta es ineficiente.
 - ❑ Cuando la **llave primaria natural se recicla**, es decir, se reutiliza o se repite con cierta periodicidad, por lo que no es única en el tiempo.

SURROGATE KEY	PRIMARY KEY
A non-natural key which aims to uniquely identify each row in the table	A minimal set of attributes (columns) in a table that uniquely identifies tuples (rows) in that table
Surrogate key is a primary key	Primary key is a candidate key
	Visit www.PEDIAA.com

No estés muy orgulloso de haber comprendido estas notas.
La habilidad para manejar el Modelo Relacional es insignificante comparado con el poder de la Fuerza.

