



# ADMINISTRADOR DE BASE DE DATOS SQL SERVER

## MODELO RELACIONAL

**Eric Gustavo Coronel Castillo**

[youtube.com/DesarrollaSoftware](https://youtube.com/DesarrollaSoftware)

[www.desarrollasoftware.com](http://www.desarrollasoftware.com)

[gcoronel@uni.edu.com](mailto:gcoronel@uni.edu.com)

# Logro Esperado

Al finalizar esta presentación, se espera que el participante entienda el modelo RELACIONAL, que es fundamental para entender las base de datos relacionales.



# Presentación

- A finales de los años sesenta Codd introdujo la teoría de las relaciones en el campo de las bases de datos.
- El documento de Codd propone un modelo de datos basado en la teoría de las relaciones, en donde los datos se estructuran lógicamente en forma de relaciones (tablas).
- El objetivo fundamental del modelo es mantener la independencia de la estructura lógica respecto al modo de almacenamiento y a otras características de tipo físico.

Código	Nombre	Edad	Semestre
D1001	Sergio Matsukawa	45	V
D1002	Hugo Valencia	38	III
D1003	Gustavo Coronel	40	VI
D1004	Ernesto Rodas	36	II

# Presentación

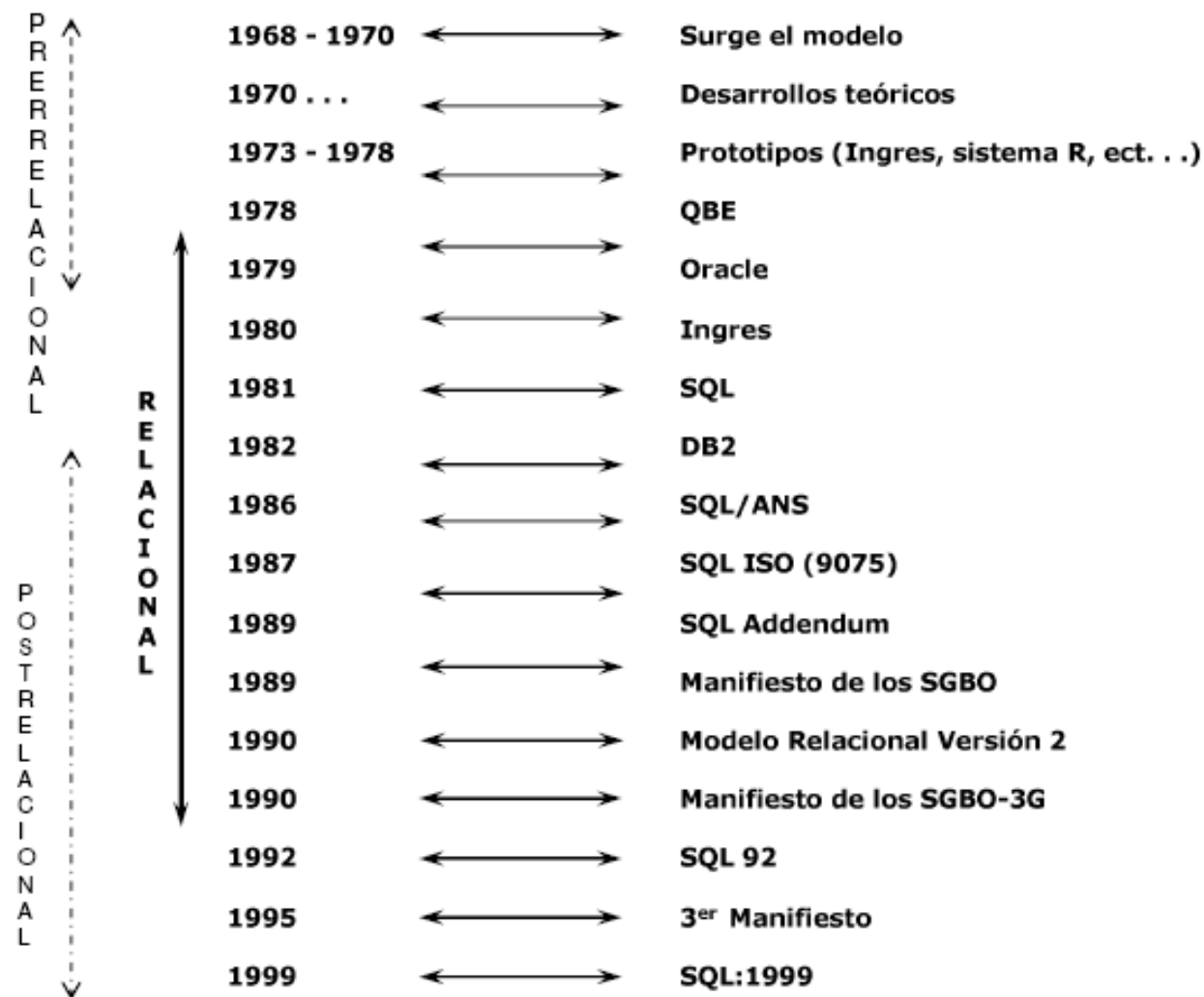
---

Los objetivos de Codd con el modelo relacional son:

- **Independencia Física:** Almacenamiento/manipulación. Un cambio en el almacenamiento físico no afecta a los programas.
- **Independencia Lógica:** Añadir, eliminar o modificar elementos en la BD no debe repercutir en los programas y/ o usuarios que acceden a ellos.
- **Flexibilidad:** Ofrecer al usuario los datos en la forma más adecuada a cada aplicación.
- **Uniformidad:** Las estructuras lógicas de los datos son tablas. Facilita la concepción y utilización de la BD por parte de los usuarios.
- **Sencillez:** Por las características anteriores y por los lenguajes de usuario sencillos, el modelo relacional es fácil de comprender y utilizar por parte del usuario final.

# Presentación

## Evolución del modelo relacional



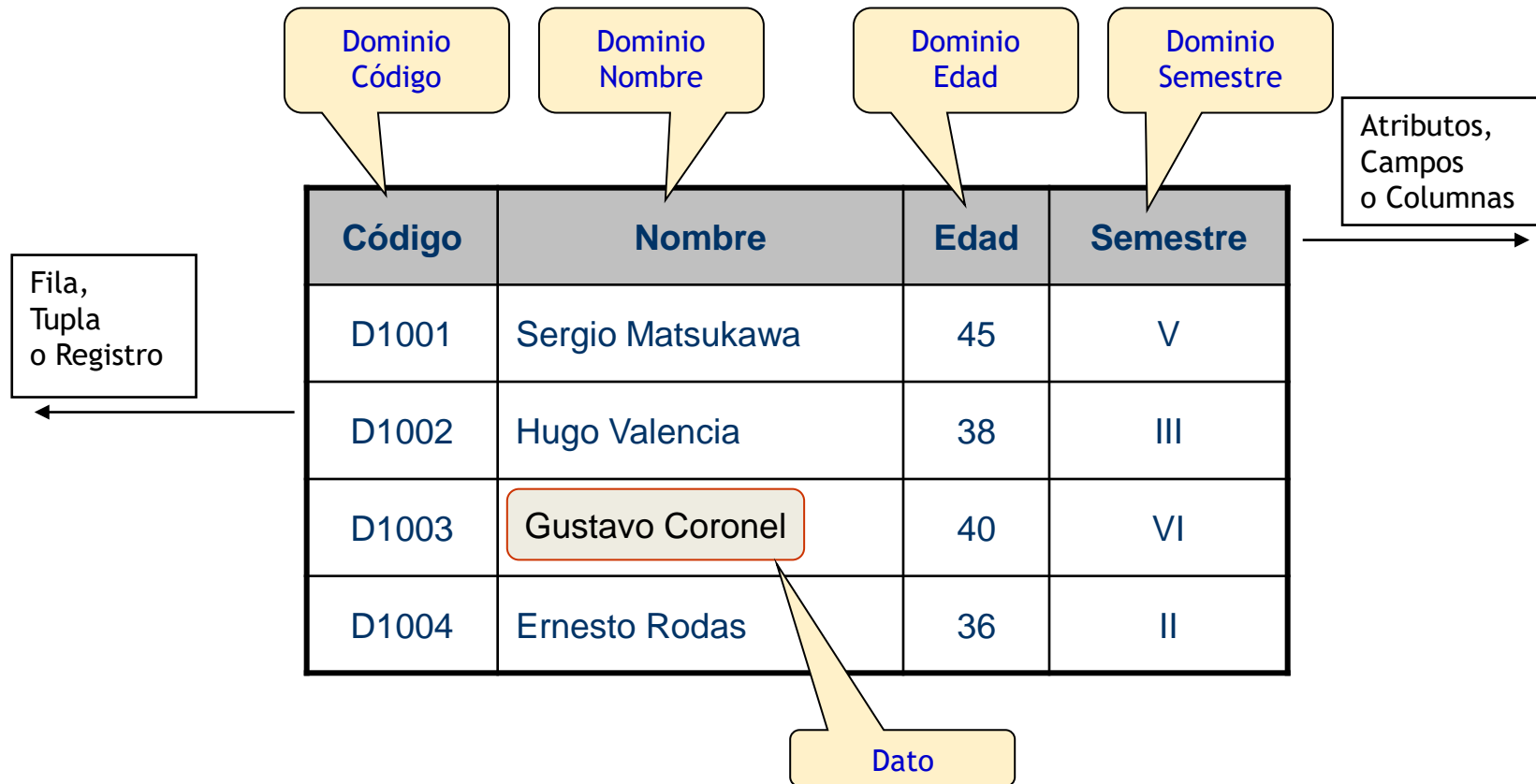
# Estructura del Modelo Relacional

---

- **Relación:** Es la estructura básica del modelo relacional. Se representa mediante una **tabla**.
- **Atributo:** Representa las propiedades de la relación. Se representa mediante una **columna**.
- **Dominio:** Es el conjunto válido de **valores** que toma un atributo.
- **Tupla:** Es una ocurrencia de la relación. Se representa mediante una **fila**.
- **Dato:** Unidad básica (atómica). Un valor del dominio.

# Estructura del Modelo Relacional

## Ejemplo de un relación:



# Estructura del Modelo Relacional

---

**Grado:** Es el número de atributos de la relación.

**Cardinalidad:** Es el número de tuplas de la relación.

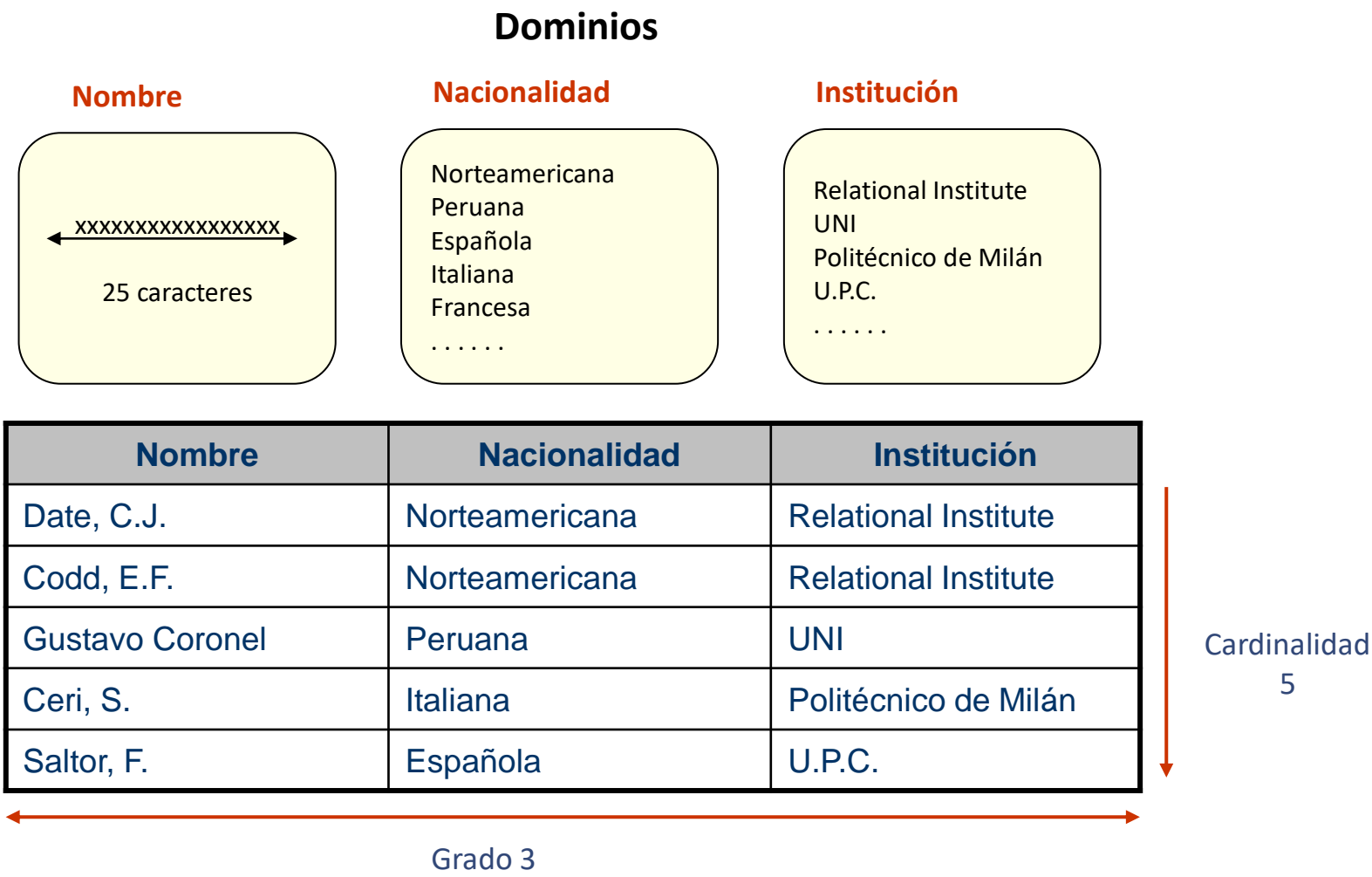
## Características de un Relación

- No puede haber tuplas duplicadas.
- El orden de las tuplas es irrelevante.
- La tabla es plana, es decir, en el cruce de un atributo y una tupla sólo puede haber un valor.
- El orden de los atributos no es significativo.



# Estructura del Modelo Relacional

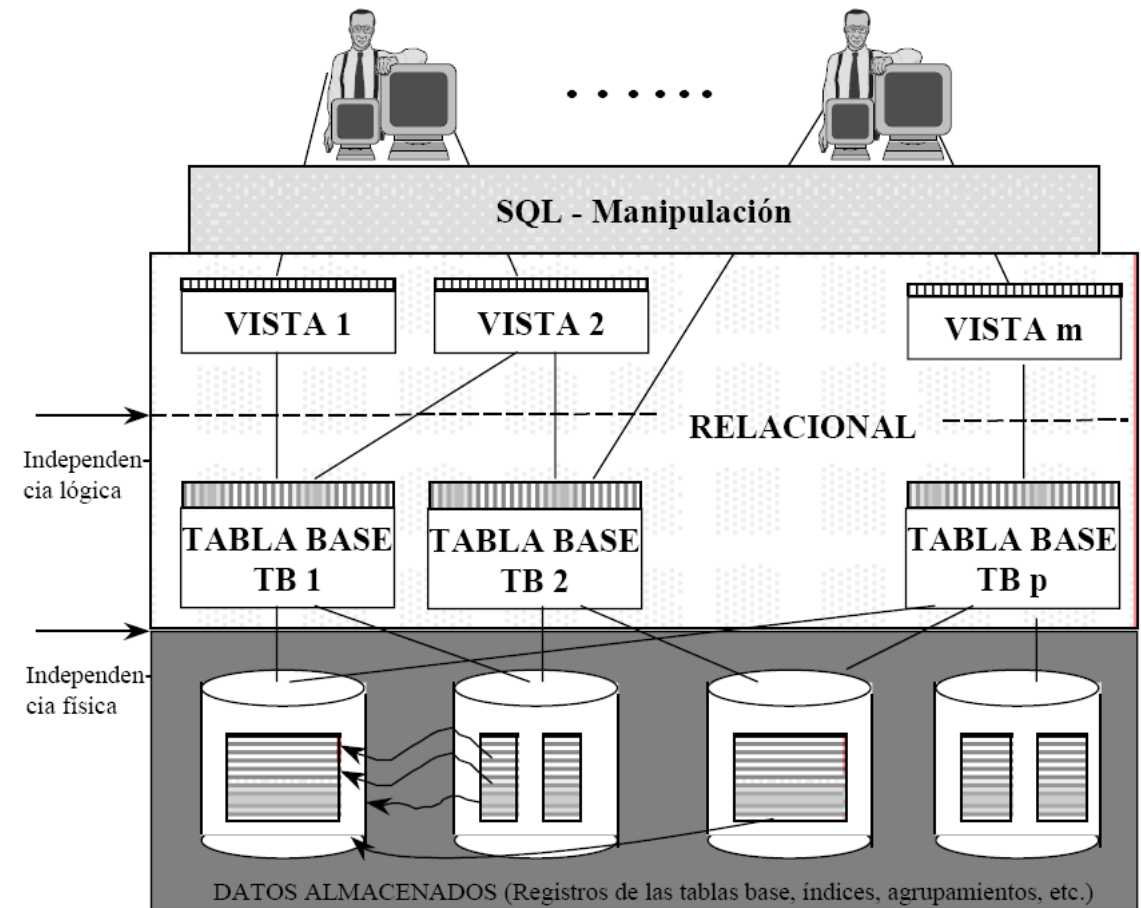
## Atributos y Dominios



# Estructura del Modelo Relacional

## Vistas

Las vistas son “tablas virtuales” que se definen sobre una o más tablas. Las vistas son ventanas sobre tablas “reales” de las que sólo se almacena su definición; no tienen representación directa en el almacenamiento.



# Definición Formal de una Relación

- Matemáticamente, una relación definida sobre los  $n$  dominios  $D_1, D_2, \dots, D_n$ , no necesariamente distintos, es un subconjunto del producto cartesiano de estos dominios, donde cada elemento de la relación, tupla, es una serie de  $n$  valores ordenados.
- Intensión o Esquema de Relación: Se compone de un nombre de relación  $R$ , de un conjunto de  $n$  atributos  $\{A_i\}$  y de un conjunto de  $n$  dominios (no necesariamente distintos)  $\{D_i\}$ , donde cada atributo será definido sobre un dominio:

$$R ( A_1 : D_1, A_2 : D_2, \dots, A_n : D_n )$$

- Extensión u Ocurrencia (instancia) de la Relación: Se llama simplemente relación, es un conjunto de  $m$  elementos denominados tuplas  $\{t_j\}$ . Cada tupla es un conjunto de  $n$  pares atributo-valor  $(\langle A_i : v_{ij} \rangle, \dots, \langle A_i : v_{ij} \rangle, \dots, \langle A_n : v_{nj} \rangle)$  donde cada  $A_i$  es el nombre de un atributo y  $v_{ij}$  es un valor del correspondiente dominio  $D_i$  sobre el que está definido el atributo:

$$r(R) = \{ t_j \mid ( \langle A_1 : v_{1j} \rangle, \dots, \langle A_i : v_{ij} \rangle, \dots, \langle A_n : v_{nj} \rangle ) : v_{ij} \in D_i \}$$

# Definición Formal de una Relación

## Intensión de una Relación

AUTOR(nombre: nombres, nacionalidad: nacionalidades, institución: instituciones)

## Extensión de una Relación

### AUTOR

Nombre	Nacionalidad	Institución
Date, C.J.	Norteamericana	Relational Institute
Codd, E.F.	Norteamericana	Relational Institute
Gustavo Coronel	Peruana	UNI
Ceri, S.	Italiana	Politécnico de Milán
Salton, F.	Española	U.P.C.

# Claves

---

## Clave Candidata

Es el conjunto no vacío de atributos que identifica **unívocamente** cada tupla de una relación.

## Clave Primaria

Es la clave candidata que elige el usuario para identificar las tuplas de la relación. Se dice que una clave primaria es **compuesta** cuando está formada por más de un atributo.

Ningún atributo principal, es decir, ningún atributo que forme parte de la clave primaria, puede tomar un valor nulo.

# Claves

---

## Claves Alternativas

Son aquellas claves candidatas que no ha sido elegidas como clave primaria.

## Clave Ajena

La clave ajena de una relación R2 es un conjunto no vacío de atributos cuyos valores han de coincidir con los valores de la clave primaria de una relación R1 (R1 y R2 no son necesariamente distintas) .

La clave ajena y la correspondiente clave primaria han de estar definidas sobre los mismos dominios.

La clave ajena sirve para relacionar tablas.

# Claves

## Representación

Tabla: Cliente

Clienteld	Nombre	Email
C1001	Sergio Matsukawa	smatsukawa@isil.edu.pe
C1002	Hugo Valencia	hvalencia@soft.com
C1003	Gustavo Coronel	perudev@hotmail.com
C1004	Ernesto Rodas	erodas@soft.com

Tabla: Pedido

Pedidoid	Fecha	Clienteld	Importe
1001	04/Ene/2006	C1002	5,000.00
1002	15/Ene/2006	C1001	10,000.00
1003	13/Feb/2006	C1004	8,500.00
1004	07/Mar/2006	C1001	7,400.00

Clienteld en la  
tabla Pedido es  
una Clave Ajena.

# Ejemplos

## Ejemplo de Relación de 1:N

LIBRO( libro\_id, titulo, idioma, editorial\_id, ... )



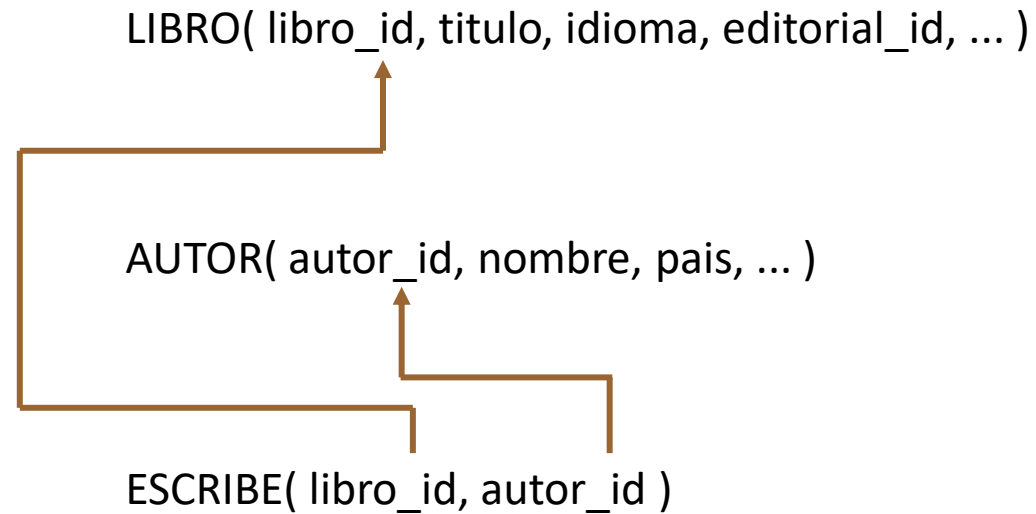
EDITORIAL( editorial\_id, direccion, ciudad, pais, ... )

El atributo **editorial\_id** es clave ajena en la entidad **LIBRO**, y referencia a la entidad **EDITORIAL** (editorial\_id es clave primaria de EDITORIAL).  
Esta última tabla se denomina *tabla referenciada*.



# Ejemplos

## Ejemplo de Relación de N:M



# Restricciones Inherentes

## Derivadas de la Definición de la Entidad

- No hay dos tuplas iguales (obligatoriedad de la clave primaria).
- El orden de las tuplas no es significativo.
- El orden de los atributos no es significativo.
- Cada atributo sólo puede tomar un único valor del dominio sobre el que está definido, no admitiéndose por tanto los grupos repetitivos.
- Se dice que una tabla que cumple esta condición está normalizada (o también que está en primera forma normal) .

## Regla de Integridad de Entidad:

Ningún atributo que forme parte de la clave primaria, puede tomar un valor nulo.

## Regla de Integridad Referencial:

Si una relación R2 tiene un atributo que es clave primaria de la relación R1, entonces los valores de dicho atributo deben concordar con los de la clave primaria o tener valores nulos.

# Restricciones Semánticas

- **Clave Primaria (PRIMARY KEY)**

Permite declarar un atributo o un conjunto de atributos como clave primaria de una relación por lo que sus valores no se podrán repetir ni se admitirán los nulos (o valores “ausentes”).

- **Unicidad (UNIQUE)**

Mediante la cual se indica que los valores de un conjunto de atributos (uno o más) no pueden repetirse en una relación. Esta restricción permite la definición de claves alternativas.

- **Obligatoriedad (NOT NULL)**

De uno o mas atributos, con lo que se indica que el conjunto de atributos no admite valores nulos.

- **Integridad Referencial (FOREING KEY)**

Si una relación R2 (relación que referencia) tiene un descriptor que es una clave candidata de la relación R1 (relación referenciada), todo valor de dicho descriptor debe, concordar con un valor de la clave candidata referenciada de R1 o ser nulo. El descriptor es, por tanto, una clave ajena de la relación R2. Las relaciones R1 y R2 no son necesariamente distintas. Además, cabe destacar que la clave ajena puede ser también parte (o la totalidad) de la clave primaria de R2.

# Restricciones

## Ejemplo

Tabla: Cliente

Clienteld	Nombre	Email
C1001	Sergio Matsukawa	smatsukawa@isil.edu.pe
C1002	Hugo Valencia	hvalencia@soft.com
C1003	Gustavo Coronel	perudev@hotmail.com
C1004	Ernesto Rodas	erodas@soft.com

Tabla: Pedido

Pedidold	Fecha	Clienteld	Importe
1001	04/Ene/2006	C1002	5,000.00
1002	15/Ene/2006	C1001	10,000.00
1003	13/Feb/2006l	C1004	8,500.00
1004	07/Mar/2006	C1001	7,400.00

Clienteld en la  
tabla Pedido es  
una Clave Ajena.

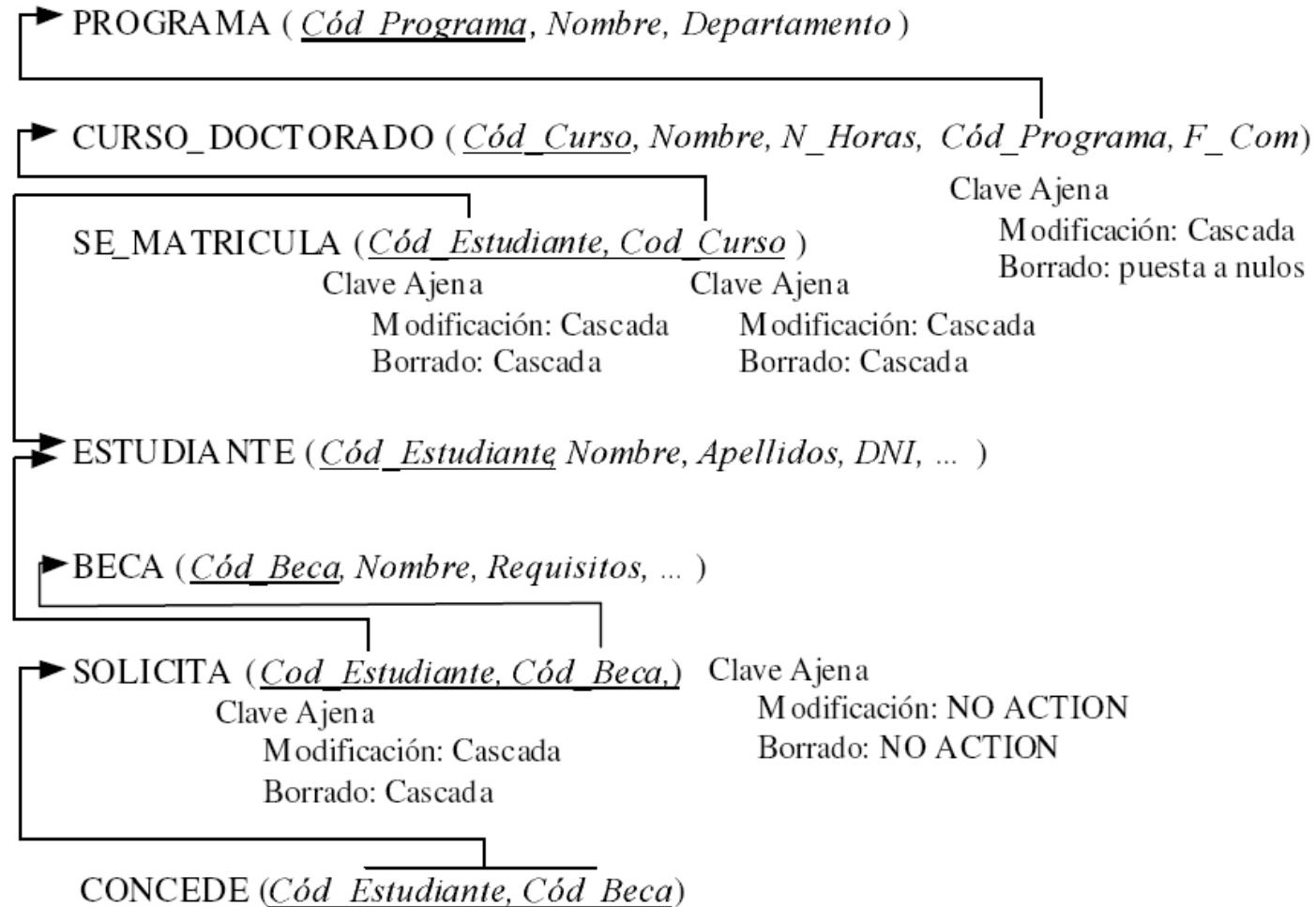
# Restricciones

## Opciones de Borrado y Actualización de la Clave Ajena

OPCIÓN	DESCRIPCIÓN
NO ACTION	Rechazar la operación de borrado o actualización.
CASCADE	Propagar la modificación o borrar las tuplas de la tabla que referencia.
SET NULL	Poner a valor nulo en la Clave Ajena (CA) de la tabla que referencia.
SET DEFAULT	Poner un valor por defecto en la CA de la tabla que referencia.

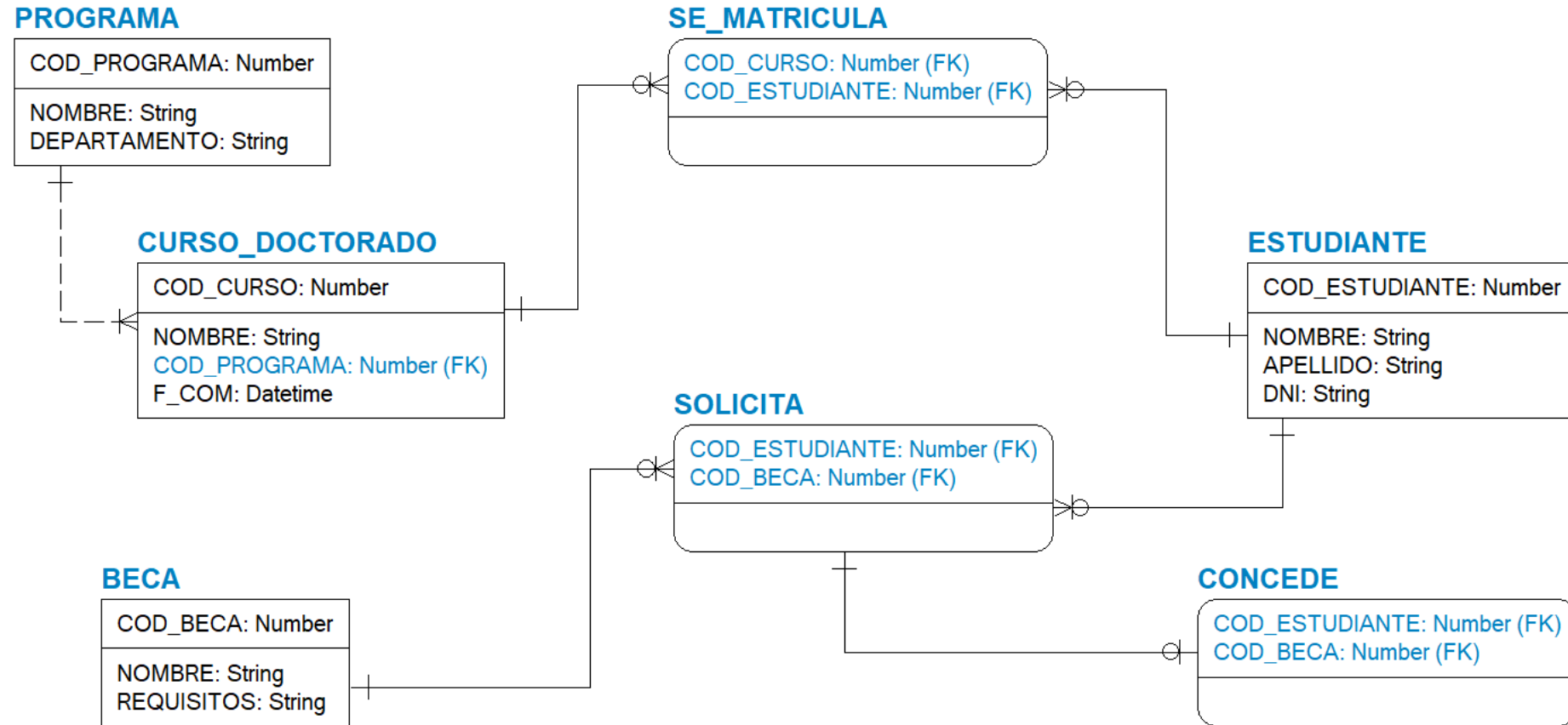
# Restricciones

## Ejemplo



# Restricciones

## Ejemplo



# Restricciones

## Otras Restricciones

- **Verificación (CHECK)**

Comprueba, en toda operación de actualización, si el predicado es cierto o falso y, en el segundo caso, rechaza la operación. La restricción de verificación se define sobre un único elemento y puede o no tener nombre.

- **Aserción (ASSERTION)**

Actúa de forma idéntica a la anterior, pero se diferencia de ella en que puede afectar a varios elementos (por ejemplo, a dos relaciones distintas) y su definición, por tanto, no va unida a la de un determinado elemento por lo que siempre ha de tener un nombre, ya que la aserción es un elemento más del esquema que tiene vida por sí mismo.

- **Disparador (trigger)**

Restricciones en las que el usuario pueda especificar libremente la respuesta (acción) ante una determinada condición. Así como las anteriores reglas de integridad son declarativas los disparadores son procedimentales, siendo preciso que el usuario escriba el procedimiento que ha de aplicarse en caso de que se cumpla la condición.



# Evaluación

---

1. ¿Cuáles son los elementos del modelo relacional?
2. ¿Puede un atributo ser clave ajena y formar parte de la clave primaria en la misma relación? Ilustre su respuesta con un ejemplo.
3. ¿Puede una clave ajena hacer referencia a la misma relación donde se encuentra definida? Ilustre su respuesta con un ejemplo.
4. ¿Cuál es el requerimiento para definir una clave ajena?
5. ¿Cuales son las restricciones inherentes de una relación?



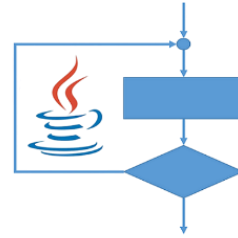
**GUSTAVO CORONEL**  
DESARROLLA SOFTWARE



**Gustavo Coronel**



[gcoronelc.github.io](https://gcoronelc.github.io)  
[youtube.com/DesarrollaSoftware](https://youtube.com/DesarrollaSoftware)  
[facebook.com/groups/desarrollasoftware](https://facebook.com/groups/desarrollasoftware)



**FUNDAMENTOS DE PROGRAMACIÓN CON JAVA**  
Inicia tu aprendizaje, utilizando las mejores prácticas de programación



**CURSO PROFESIONAL DE JAVA ORIENTADA A OBJETOS**  
Aprende programación en capas, patrones y buenas prácticas



**PROGRAMACIÓN DE BASE DE DATOS ORACLE CON PL/SQL**  
Aprende a obtener el mejor rendimiento de tú base de datos



**PROGRAMACIÓN DE BASE DE DATOS ORACLE CON JDBC**  
Aprende a programar correctamente con JDBC