NORMALIZACIÓN
Parte 1

Eric Gustavo Coronel Castillo

youtube.com/DesarrollaSoftware

www.desarrollasoftware.com

gcoronel@uni.edu.com

Logro Esperado

Al finalizar esta presentación, se espera que el participante entienda la importancia del concepto de **dependencia funcional**, que es la base para entender y aplicar la normalización.

Índice

- Necesidad de un Método Formal en el Diseño Relacional
- Redundancia de Datos
- La Necesidad de Normalizar
- Las Formas Normales
- Dependencia Funcional
- Ejercicios

NECESIDAD DE UN MÉTODO FORMAL EN EL DISEÑO RELACIONAL

Las relaciones que resultan de la observación del mundo real o de la transformación del esquema ER elaborado en la etapa del modelo conceptual, pueden presentar algunos problemas, derivados de fallas en la percepción del sistema, en el diseño del esquema ER o en el paso al modelo relacional; entre estos problemas cabe destacar los siguientes:

- Incapacidad para almacenar ciertos hechos.
- Redundancia, y por lo tanto, posibilidad de incoherencias.
- Ambigüedades.
- Pérdida de información (aparición de tuplas huérfanas)
- Pérdida de dependencias funcionales, es decir, de ciertas restricciones de integridad que dan lugar a interdependencias entre los datos.
- Aparición, en la base de datos, de estados que no son válidos en el mundo real, es decir, anomalías de inserción, borrado y modificación.

NECESIDAD DE UN MÉTODO FORMAL EN EL DISEÑO RELACIONAL

Ejemplo de un modelo inadecuado

Autor	Nacionalidad	CodLibro	Titulo	Editorial	Año
Date, C.	Norteamérica	98987	Databases	Addison-W	1990
Date, C.	Norteamérica	97777	SQL Standard	Addison-W	1986
Date, C.	Norteamérica	98987	A Guide to Ingres	Addison-W	1988
Codd, E.	Norteamérica	7890	Relation Model	Addison-W	1990
Gardarin	Francesa	12345	Base de Datos	Paraninfo	1986
Gardarin	Francesa	67890	Comparación BD	Enrolles	1984
Valduriez	Francesa	67890	Comparación BD	Eyrolles	1984
Kim, W.	Norteamérica	11223	O-O Database	ACM Press	1989
Lochovsky	Canadiense	11223	O-O Database	ACM Press	1989

NECESIDAD DE UN MÉTODO FORMAL EN EL DISEÑO RELACIONAL

En este modelo observamos:

- Gran cantidad de redundancia; ya que la nacionalidad del autor se repite por cada ocurrencia del mismo, y algo análogo sucede cuando un libro tiene más de un autor, con la editorial y el año de publicación.
- Anomalías de modificación; ya que, inadvertidamente podemos, por ejemplo, cambiar el nombre de la editorial en una tupla sin modificarlo en el resto de las que corresponden al mismo libro, lo que da lugar a incoherencias.
- Anomalías de inserción; ya que si se quisiera incluir información sobre algún autor del que no hubiera ningún libro en la base de datos, no sería posible, al formar el atributo cod_libro parte de la clave primaria de la relación, tampoco podríamos introducir obras anónimas. Además, la inserción de un libro que tuviera dos autores obligaría a incluir dos tuplas en la base de datos.
- Anomalías de borrado, ya que si quisiéramos dar de baja un libro, también se perderían datos sobre sus autores y, viceversa.

NECESIDAD DE UN MÉTODO FORMAL EN EL DISEÑO RELACIONAL

- Esta relación presenta todos estos problemas, y alguno más, debido a que atenta contra un principio básico en todo diseño:

"hechos distintos se deben almacenar en objetos distintos"

- En este caso, en relaciones distintas, con lo cual se habrían evitado redundancia, y por tanto, anomalías de actualización.

REDUNDANCIA DE DATOS

- Redundancia significa la repetición de datos.
- La redundancia incrementa el tiempo que se ocupa para actualizar, agregar y eliminar datos.
- La redundancia también incrementa el espacio que se utiliza en el disco y por ende, aumentan las lecturas y escrituras.
- La redundancia puede causar:
  - Anomalías en las actualizaciones: la inserción, modificación y eliminación de datos pueden causar inconsistencias.
  - Inconsistencias: es más probable que surjan errores cuando se repiten los hechos.
  - La utilización innecesaria de espacio extra en el disco.

Redundancia de Datos

Alumno
AluID
AluNombre
AluFecNac
AluDireccion
AluCiudad
AluCarrera
AluSemestre
AluNota1
AluNota2

AluID	AluNombre	. . .	AluSemestre	AluNota1	AluNota2
001	Mary	. . .	Sem-1	15	12
001	Mary	. . .	Sem-2	13	16
002	Jake	. . .	Sem-1	13	15
002	Jake	. . .	Sem-2	10	14
002	Jake	. . .	Sem-2	18	11
003	Peter	. . .	Sem-1	14	12
003	Peter	. . .	Sem-2	17	15
004	Scott	. . .	Sem-1	16	14
004	Scott	. . .	Sem-2	17	12

- Los detalles de los estudiantes como **AluID**, **AluNombre**, y **AluDireccion** son repetidos mientras se registra las notas de diferentes semestres.
- Los datos repetidos son redundantes.
- Si se necesita modificar la dirección de un alumno, se tienen que modificar múltiples filas para ese alumno. Si no, podríamos tener inconsistencias a través de las filas.
- Si hay mil alumnos y los detalles para cada alumno ocupan 200 bytes, entonces tenemos 200,000 bytes que se repiten. Por eso, mucho espacio de disco es usado innecesariamente.

LA NECESIDAD DE NORMALIZAR

- La normalización es el método científico que divide las estructuras complejas de las tablas en estructuras de tablas sencillas, con el uso de ciertas reglas.
- Al usar la normalización, usted puede reducir la redundancia en una tabla y eliminar los problemas de inconsistencia y el uso de espacio en el disco.
- También puede asegurar que no se pierda información.
- La normalización ofrece los siguientes beneficios:
  - Acelera el ordenamiento y la creación de índices
  - Ayudar a crear más índices agrupados
  - Cada tabla requiere menos índices
  - Reduce el número de valores NULOS en una tabla
  - Compacta la base de datos

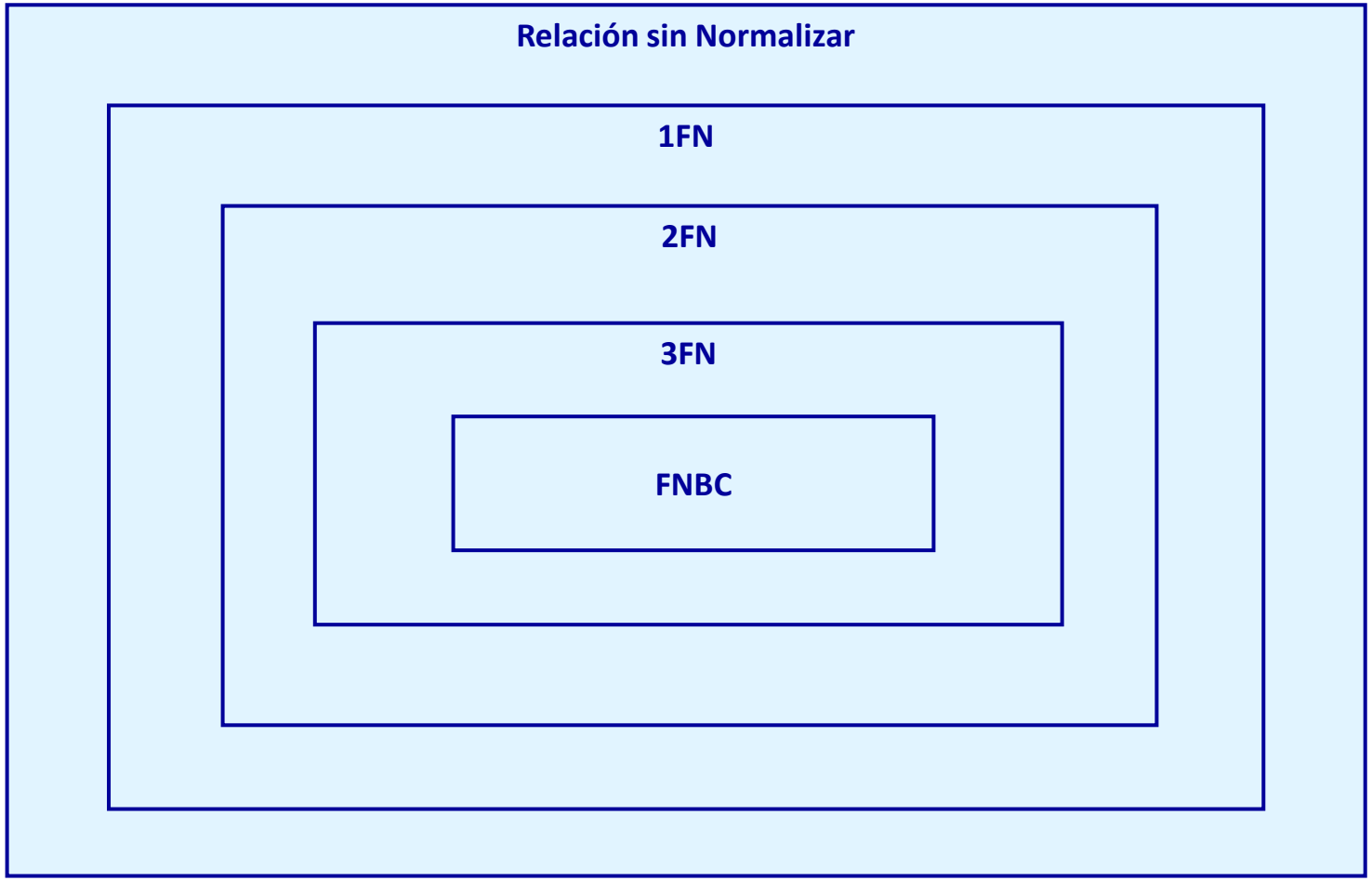
LA NECESIDAD DE NORMALIZAR

- El desempeño de una aplicación está directamente relacionado con el diseño de la base de datos.
- Se deben seguir las siguientes reglas para diseñar una buena base de datos:
  - Cada tabla debe tener un identificador
  - Cada tabla debe guardar datos para un entidad única
  - Se deben evitar las columnas que aceptan valores NULOS
  - Se debe evitar la repetición de valores o de columnas

LAS FORMAS NORMALES

- La normalización sirve para formar tablas que cumplen con ciertas reglas específicas y representan ciertos formas normales.
- Los formas normales se utilizan para asegurar que no se introduzcan en la base de datos varios tipos de anomalías e inconsistencias.
- La estructura de una tabla siempre aparece en un cierta forma normal.
- Se han identificado varias formas normales. Los formas normales más importantes y populares son:
  - La Primera Forma Normal (1FN)
  - La Segunda Forma Normal (2FN)
  - La Tercera Forma Normal (3FN)
  - La Forma Normal de Boyce-Codd (FNBC)

Las Formas Normales



DEPENDENCIA FUNCIONAL

- La teoría de la normalización se basa en el concepto de dependencias, hasta el punto que actualmente se conoce también como teoría de las dependencias.
- La existencia de una dependencia no se puede demostrar, pero si afirmar por observación del mundo real que se trata de modelar.
- Las dependencias nos muestra algunas importantes interrelaciones existentes entre los atributos del mundo real, cuya semántica tratamos de incorporar a nuestra base de datos; son, por lo tanto, invariantes en el tiempo, siempre que no cambie el mundo real del cual proceden.
- En el proceso de normalización será fundamental empezar identificando todas la dependencias funcionales del universo del caso cuyo diseño estamos realizando, y preocuparemos conservar las dependencias funcionales a los largo del proceso, de modo que el esquema relacional resultante de la normalización tenga las mismas dependencias funcionales que el esquema de partida, o un conjunto equivalente.

Dependencia Funcional

- En una relación **R**, el atributo **B** depende funcionalmente del atributo **A** si cada valor de **B** en **R** está asociado exactamente con un valor de **A**.
- Al atributo **A** se le llama determinante.
- Todos los atributos de una tabla deben depender funcionalmente de la clave. Sin embargo, la dependencia funcional no necesita que un atributo sea una clave para determinar funcionalmente a otros atributos.
- La dependencia funcional también se puede definir de la siguiente manera:

  Dada una relación R, el atributo B depende funcionalmente de A únicamente cuando dos tuplas R coinciden en su valor A, deben coincidir en su valor B.

Dependencia Funcional

Tabla: Empleado

Code	Name	City
E1	Claudia	Chiclayo
E2	Sandra	Trujillo
E3	Henry	Cuzco

Code → Name

Code → City

Code → Name, City

- Dado un valor particular de **Code**, hay exactamente un valor correspondiente para **Name**.
- Por ejemplo, para Code **E1** hay exactamente un valor de Name, **Claudia**. Por eso **Name** es funcionalmente dependiente de **Code**.
- De manera similar, existe exactamente un valor de City para cada valor de Code. Por eso el atributo **City** es funcionalmente dependiente de **Code**.
- El atributo **Code** es determinante.
- También se puede decir que Code determina City y Name.

Dependencia Funcional

Tabla: Notas

ID	City	C_Code	Nota
AD0036	London	C1	15
AD0078	New York	C1	16
CC0075	New York	C2	18
CC0097	Florida	C1	13
AD0036	London	C2	15
CC0075	New York	C1	14

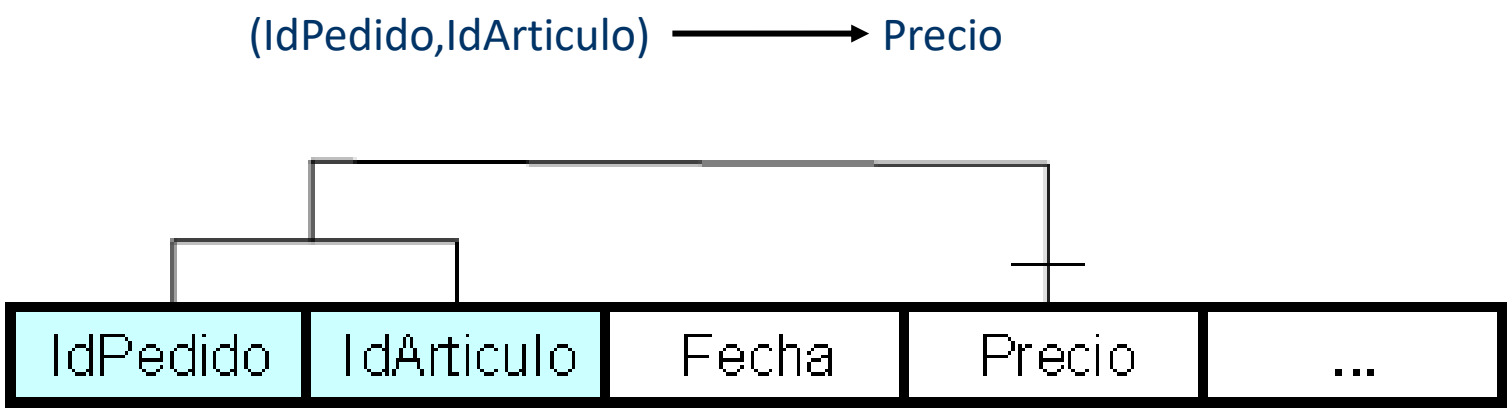
Tabla de notas de alumnos del programa de educación a distancia.

- El ID determina funcionalmente la CITY.
- Pero el ID no es candidato a clave. El candidato a clave en este caso, será la combinación de ID y C_CODE.
- Los atributos ID y C_CODE son claves foráneas que hacen referencia a tablas que almacenan información del alumno y el curso respectivamente.
- Por lo tanto, aunque ID no es candidato a clave, determina funcionalmente otro atributo (CITY).
- Note que para un valor particular de ID, el valor de CITY es el mismo en cualquier fila.
- Que ID determina funcionalmente la CITY significa que cada alumno esta localizado en una ciudad.
- Esta dependencia funcional (ID determina CITY) también significa que muchos estudiantes están localizados en una ciudad, pero un estudiante esta localizado solo en una ciudad.

Dependencia Funcional

Dependencia Funcional Completa

Se dice que B tiene dependencia funcional completa o plena de A, si depende funcionalmente de A, pero no depende de ningún subconjunto de A.



Dependencia Funcional

Descriptores Equivalentes

- Se dice que dos descriptores X e Y son equivalentes, si se cumple que:

$$X \rightarrow Y$$

$$Y \rightarrow X$$

- Lo cual se puede representar por:

$$X \leftrightarrow Y$$

- Por Ejemplo:

$$IdArticulo \leftrightarrow NombArticulo$$

Dependencia Funcional

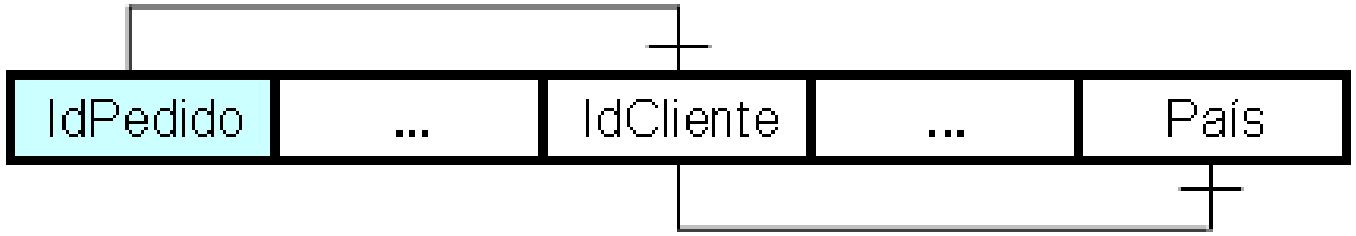
Dependencia Funcional Transitiva

Se dice que C tiene dependencia transitiva respecto de A, a través de B si:

$$A \rightarrow B$$
$$B \rightarrow C$$

Por ejemplo:

IdPedido → IdCliente
IdCliente → País



Ejercicios

Para los siguientes casos, por separado determine su diagrama relacional y sus dependencias funcionales.

1. En la empresa **ABC** se necesita un documento para que los empleados puedan hacer las solicitudes de sus requerimientos de útiles de escritorio para el desarrollo de sus labores.

2. El gerente de **Cines Perú** necesita un documento para registrar la cantidad de personas que han ingresado en cada función de su salas de cine, cuenta con 10 salas, y en una sala se proyectan un promedio de 6 películas por día, depende del tiempo de duración de cada película.

GUSTAVO CORONEL

DESARROLLA SOFTWARE

Eric Gustavo Coronel Castillo

www.desarrollasoftware.com

INSTRUCTOR

Udemy

https://github.com/gcoronelc/UDEMY

FUNDAMENTOS DE PROGRAMACIÓN CON JAVA

Aprende las mejores prácticas

PROGRAMACIÓN ORIENTADA A OBJETOS CON JAVA

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