

Normalización

30/Agosto/2019



Normalización - Marco General

- **Salida del Diseño.** Conjunto de relaciones

Normalización - Marco General

- **Salida del Diseño.** Conjunto de relaciones
- **Calidad de Diseño.** Necesidad de evaluar si una forma de agrupar atributos en un esquema es mejor que otra
- **Niveles.**
 - 1 **Lógico (o Conceptual).** Un buen diseño de esquemas a este nivel habilita a los usuarios a entender el significado de los datos de las relaciones
 - 2 **Implementación (o de Almacenamiento Físico).** Cómo se almacenan y actualizan las tuplas

Normalización - Marco General

- **Salida del Diseño.** Conjunto de relaciones
- **Calidad de Diseño.** Necesidad de evaluar si una forma de agrupar atributos en un esquema es mejor que otra
- **Niveles.**
 - ① **Lógico (o Conceptual).** Un buen diseño de esquemas a este nivel habilita a los usuarios a entender el significado de los datos de las relaciones
 - ② **Implementación (o de Almacenamiento Físico).** Cómo se almacenan y actualizan las tuplas
- **Objetivos.**
 - **Preservar la Información.** Conceptos
 - **Minimizar Redundancia** Evitar almacenamiento de información redundante

Normalización - Marco General

- **Salida del Diseño.** Conjunto de relaciones
- **Calidad de Diseño.** Necesidad de evaluar si una forma de agrupar atributos en un esquema es mejor que otra
- **Niveles.**
 - ① **Lógico (o Conceptual).** Un buen diseño de esquemas a este nivel habilita a los usuarios a entender el significado de los datos de las relaciones
 - ② **Implementación (o de Almacenamiento Físico).** Cómo se almacenan y actualizan las tuplas
- **Objetivos.**
 - **Preservar la Información.** Conceptos
 - **Minimizar Redundancia** Evitar almacenamiento de información redundante
- **Pautas de Diseño.** Cuatro pautas informales de diseño pueden utilizarse como medida para determinar la calidad de un diseño:

Normalización - Marco General

- **Salida del Diseño.** Conjunto de relaciones
- **Calidad de Diseño.** Necesidad de evaluar si una forma de agrupar atributos en un esquema es mejor que otra
- **Niveles.**
 - 1 **Lógico (o Conceptual).** Un buen diseño de esquemas a este nivel habilita a los usuarios a entender el significado de los datos de las relaciones
 - 2 **Implementación (o de Almacenamiento Físico).** Cómo se almacenan y actualizan las tuplas
- **Objetivos.**
 - **Preservar la Información.** Conceptos
 - **Minimizar Redundancia** Evitar almacenamiento de información redundante
- **Pautas de Diseño.** Cuatro pautas informales de diseño pueden utilizarse como medida para determinar la calidad de un diseño:
 - 1 Estar seguro que semántica de atributos en esquemas es clara
 - 2 Reducir la información redundante en tuplas
 - 3 Reducir la cantidad de valores NULL en tuplas
 - 4 Desabilitar la posibilidad de generar tuplas espúreas
- **Independencia.** Estas pautas NO son siempre independientes unas de otras

Normalización - Pauta Nro. 1 - Semántica

- **Semántica.** Cuanto más fácil es explicar la semántica de los esquemas, mejor es el diseño.

Normalización - Pauta Nro. 1 - Semántica

- **Semántica.** Cuanto más fácil es explicar la semántica de los esquemas, mejor es el diseño.
- **Ejemplo.**

Normalización - Pauta Nro. 1 - Semántica

- **Semántica.** Cuanto más fácil es explicar la semántica de los esquemas, mejor es el diseño.
- **Ejemplo.**

EMPLEADO_PROYECTO

E_Nombre	<u>E_DNI</u>	E_Fecha_Nacimiento	Dirección	P_Nombre	<u>P_Número</u>
----------	--------------	--------------------	-----------	----------	-----------------

Normalización - Pauta Nro. 1 - Semántica

- **Semántica.** Cuanto más fácil es explicar la semántica de los esquemas, mejor es el diseño.
- **Ejemplo.**

EMPLEADO_PROYECTO

E_Nombre	<u>E_DNI</u>	E_Fecha_Nacimiento	Dirección	P_Nombre	<u>P_Número</u>
----------	--------------	--------------------	-----------	----------	-----------------

¿Opinión?

Normalización - Pauta Nro. 1 - Semántica

- **Semántica.** Cuanto más fácil es explicar la semántica de los esquemas, mejor es el diseño.
- **Ejemplo.** EMPLEADO_PROYECTO

E_Nombre	<u>E_DNI</u>	E_Fecha_Nacimiento	Dirección	P_Nombre	<u>P_Número</u>
----------	--------------	--------------------	-----------	----------	-----------------

¿Opinión?

- Mezcla atributos de EMPLEADO con PROYECTO
- Desde el punto de vista de la lógica, puede ser correcto
- Deficiente en cuanto a la calidad respecto de la semántica de la relación (Pauta Nro. 1)
- Puede ser utilizado como vista

Normalización - Pauta Nro. 1 - Semántica

- **Semántica.** Cuanto más fácil es explicar la semántica de los esquemas, mejor es el diseño.

- **Ejemplo.**

EMPLEADO_PROYECTO

E_Nombre	<u>E_DNI</u>	E.Fecha_Nacimiento	Dirección	P_Nombre	<u>P_Número</u>
----------	--------------	--------------------	-----------	----------	-----------------

¿Opinión?

- Mezcla atributos de EMPLEADO con PROYECTO
- Desde el punto de vista de la lógica, puede ser correcto
- Deficiente en cuanto a la calidad respecto de la semántica de la relación (Pauta Nro. 1)
- Puede ser utilizado como vista

- **Ejemplo OK.**

EMPLEADO

<u>E_Nombre</u>	<u>E_DNI</u>	E.Fecha_Nacimiento	Dirección_Laboral
-----------------	--------------	--------------------	-------------------

PROYECTO

<u>P_Nombre</u>	<u>P_Número</u>
-----------------	-----------------

TRABAJA_EN

<u>E_DNI</u>	<u>P_Número</u>
--------------	-----------------

Normalización - Pauta Nro. 1 - Semántica

- **Semántica.** Cuanto más fácil es explicar la semántica de los esquemas, mejor es el diseño.
- **Ejemplo.** EMPLEADO_PROYECTO

<u>E_Nombre</u>	<u>E_DNI</u>	<u>E.Fecha_Nacimiento</u>	<u>Dirección</u>	<u>P_Nombre</u>	<u>P_Número</u>
-----------------	--------------	---------------------------	------------------	-----------------	-----------------

¿Opinión?

- Mezcla atributos de EMPLEADO con PROYECTO
- Desde el punto de vista de la lógica, puede ser correcto
- Deficiente en cuanto a la calidad respecto de la semántica de la relación (Pauta Nro. 1)
- Puede ser utilizado como vista
- **Ejemplo OK.** EMPLEADO

<u>E_Nombre</u>	<u>E_DNI</u>	<u>E.Fecha_Nacimiento</u>	<u>Dirección_Laboral</u>
-----------------	--------------	---------------------------	--------------------------

PROYECTO

<u>P_Nombre</u>	<u>P_Número</u>
-----------------	-----------------

TRABAJA_EN

<u>E_DNI</u>	<u>P_Número</u>
--------------	-----------------

- **Pauta Nro. 1**
 - Diseñar esquemas tal que sea fácil de explicar su significado
 - No combinar atributos de diversos tipos de entidades y relaciones en una misma relación

Normalización - Pauta Nro. 2 - Almacenamiento

- **Objetivo.** Minimizar espacio de almacenamiento a través del diseño

Normalización - Pauta Nro. 2 - Almacenamiento

- **Objetivo.** Minimizar espacio de almacenamiento a través del diseño
- **Ejemplo.** ¿Qué hacer para que este diseño ocupe menos espacio de almacenamiento?

Diseño "A"

EMPLEADO_DEPARTAMENTO

E_Nombre	E_DNI	E_Fecha_Nacimiento	Nro_Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
...

Normalización - Pauta Nro. 2 - Almacenamiento

- **Objetivo.** Minimizar espacio de almacenamiento a través del diseño
- **Ejemplo.** ¿Qué hacer para que este diseño ocupe menos espacio de almacenamiento?

Diseño "A"

EMPLEADO DEPARTAMENTO

E.Nombre	E.DNI	E.Fecha_Nacimiento	Nro_Depto	D.Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
...

Diseño "B"

EMPLEADO

E.Nombre	E.DNI	E.Fecha_Nacimiento	Nro_Depto
Diego	20222333	11/12/1970	5
Laura	33456234	02/04/1985	5
Marina	45432345	23/07/2006	2
Santiago	24345345	18/02/1975	5
...

DEPARTAMENTO

Nro_Depto	D.Nombre
5	Publicidad y Promoción
2	Reclutamiento y Selección

Normalización - Pauta Nro. 2 - Almacenamiento

- **Objetivo.** Minimizar espacio de almacenamiento a través del diseño
- **Ejemplo.** ¿Qué hacer para que este diseño ocupe menos espacio de almacenamiento?

Diseño "A"

EMPLEADO DEPARTAMENTO

E.Nombre	E.DNI	E.Fecha_Nacimiento	Nro_Depto	D.Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
...

Diseño "B"

EMPLEADO

E.Nombre	E.DNI	E.Fecha_Nacimiento	Nro_Depto
Diego	20222333	11/12/1970	5
Laura	33456234	02/04/1985	5
Marina	45432345	23/07/2006	2
Santiago	24345345	18/02/1975	5
...

DEPARTAMENTO

Nro_Depto	D.Nombre
5	Publicidad y Promoción
2	Reclutamiento y Selección

- **Diseño "A"** almacena NATURAL JOIN de **Diseño "B"**.
- **Anomalías de Actualización.** Almacenar NATURAL JOINS introduce problemas adicionales. **Anomalías.** Inserción, Delección y Modificación.

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Inserción.

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
...

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Inserción.

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
...

- ¿Qué sucede si se desea insertar un nuevo empleado y se desconoce ó aún no ha sido asignado a un Departamento?

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Inserción.

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
...

- ¿Qué sucede si se desea insertar un nuevo empleado y se desconoce ó aún no ha sido asignado a un Departamento?

Insertar nuevo empleado requiere incluir valores en atributos de departamento o NULL (si aún no ha sido asignado a ninguno)

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_Depto	D_Nombre
...
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
Tamara	27354632	28/02/1979	NULL	NULL

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Inserción.

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
...

- ¿Qué sucede si se desea insertar un nuevo empleado y se desconoce ó aún no ha sido asignado a un Departamento?

Insertar nuevo empleado requiere incluir valores en atributos de departamento o NULL (si aún no ha sido asignado a ninguno)

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_Depto	D_Nombre
...
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
Tamara	27354632	28/02/1979	NULL	NULL

- ¿Qué problema surge al insertar empleado asociado al Depto. 5?

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Inserción.

E.Nombre	E.DNI	E.Fecha_Nacimiento	Nro_Depto	D.Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
...

- ¿Qué sucede si se desea insertar un nuevo empleado y se desconoce ó aún no ha sido asignado a un Departamento?

Insertar nuevo empleado requiere incluir valores en atributos de departamento o NULL (si aún no ha sido asignado a ninguno)

E.Nombre	E.DNI	E.Fecha_Nacimiento	Nro_Depto	D.Nombre
...
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
Tamara	27354632	28/02/1979	NULL	NULL

- ¿Qué problema surge al insertar empleado asociado al Depto. 5?

Insertar nuevo empleado a departamento 5, requiere que los datos del departamento sean *consistentes* con el resto de los registros

E.Nombre	E.DNI	E.Fecha_Nacimiento	Nro_Depto	D.Nombre
...
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
Tamara	27354632	28/02/1979	5	Publicaciones y Prop.

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Inserción.

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro.Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
...

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Inserción.

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro.Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
...

- ¿Es posible insertar un nuevo departamento que aún no posee empleados asignados?

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Inserción.

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro.Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
...

- ¿Es posible insertar un nuevo departamento que aún no posee empleados asignados? **¡No!**

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Inserción.

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro.Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
...

- ¿Es posible insertar un nuevo departamento que aún no posee empleados asignados? **¡No!**
 - 1 NULL en campos de empleados viola la integridad de la entidad (NULL en atributo clave E_DNI)
 - 2 Cuando se asigna el primer empleado a dicho depto. esta tupla ya no es mas necesaria.

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Delección.

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Delección.

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción

- ¿Qué consecuencia tiene eliminar el registro correspondiente a Marina?

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Delección.

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción

- ¿Qué consecuencia tiene eliminar el registro correspondiente a Marina?

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Santiago	24345345	18/02/1975	5	Publicidad y Promoción

Se pierde toda la información correspondiente al departamento 2

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Modificación.

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Modificación.

E_Nombre	E_DNI	E_Fecha_Nacimiento	Nro_Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción

- ¿Qué sucede si se desea modificar “Publicidad y Promoción” por “Publicidad, Promoción y Comunicación Integral”

Normalización - Pauta Nro. 2 - Almacenamiento

1. Anomalías de Modificación.

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad y Promoción
Laura	33456234	02/04/1985	5	Publicidad y Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad y Promoción

- ¿Qué sucede si se desea modificar “Publicidad y Promoción” por “Publicidad, Promoción y Comunicación Integral”
 Modificar el valor de un atributo de un departamento requiere modificar **TODAS** las tuplas de ese departamento. Caso contrario, se generan **inconsistencias**.

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_Depto	D_Nombre
Diego	20222333	11/12/1970	5	Publicidad, Promoción y Comunicación Integral
Laura	33456234	02/04/1985	5	Publicidad, Promoción
Marina	45432345	23/07/2006	2	Reclutamiento y Selección
Santiago	24345345	18/02/1975	5	Publicidad, Promoción

Normalización - Pauta Nro. 2 - Almacenamiento

● Pauta Nro. 2.

- Diseñar esquemas tal que no permitan anomalías de inserción, deleción y modificación
- Si permiten anomalías, señalarlas claramente y asegurar que programas que actualizan BD operarán correctamente

Normalización - Pauta Nro. 2 - Almacenamiento

● Pauta Nro. 2.

- Diseñar esquemas tal que no permitan anomalías de inserción, delección y modificación
- Si permiten anomalías, señalarlas claramente y asegurar que programas que actualizan BD operarán correctamente

● Performance.

- Notar que esta pauta puede ser violada en favor de la performance
- Ejemplo. Guardar en cada factura cuánto falta pagar (saldo). Esto claramente se puede recuperar “recorriendo” los pagos asociados a una factura, pero hay que hacerlo cada vez que un usuario pregunta cuánto debe un cliente determinado, y es una pregunta bastante frecuente. El costo de esto es que, cada vez que se paga una factura, o se anula un pago hay que ir a actualizar ese número
- En tal caso se debe señalar y actuar en consecuencia (Ej. triggers/store procedures que realicen automáticamente actualizaciones)

Normalización - Pauta Nro. 3 - NULLs

- **Esquemas.** Atributos no relacionados y agrupados en una misma tabla pueden generar múltiples NULLs en una misma tupla.

Normalización - Pauta Nro. 3 - NULLs

- **Esquemas.** Atributos no relacionados y agrupados en una misma tabla pueden generar múltiples NULLs en una misma tupla.
- **Ejemplo.**

EMPLEADO_DEPARTAMENTO

E_Nombre	E_DNI	E_Fecha_Nacimiento	Nro_Depto	D_Nombre
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
Tamara	27354632	28/02/1979	NULL	NULL

Normalización - Pauta Nro. 3 - NULLs

- **Esquemas.** Atributos no relacionados y agrupados en una misma tabla pueden generar múltiples NULLs en una misma tupla.

- **Ejemplo.**

EMPLEADO_DEPARTAMENTO

E_Nombre	E_DNI	E_Fecha_Nacimiento	Nro_Depto	D_Nombre
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
Tamara	27354632	28/02/1979	NULL	NULL

- **Problemas.** ¿Qué sucede en cuanto espacio, semántica, JOIN?

Normalización - Pauta Nro. 3 - NULLs

- **Esquemas.** Atributos no relacionados y agrupados en una misma tabla pueden generar múltiples NULLs en una misma tupla.
- **Ejemplo.**

EMPLEADO_DEPARTAMENTO

E_Nombre	E_DNI	E_Fecha_Nacimiento	Nro_Depto	D_Nombre
Santiago	24345345	18/02/1975	5	Publicidad y Promoción
Tamara	27354632	28/02/1979	NULL	NULL

- **Problemas.** ¿Qué sucede en cuanto espacio, semántica, JOIN?
 - Desperdicio espacio almacenamiento
 - JOINS (en presencia de NULLs, INNER JOIN produce distinto resultado vs. OUTER JOIN)
 - ¿Cómo se interpretan funciones de agregación (COUNT, SUM, etc.)?
 - Diversas interpretaciones de NULL
 - El resultado no aplica a la tupla. Ej. Registro_Conducir no aplica a menores
 - Valor conocido pero ausente. Ej. Fecha_Nacimiento de un empleado puede ser desconocida
 - Valor desconocido (No sabemos si existe). Ej. Un empleado puede que tenga telefono y no sabemos el valor o puede que no tenga.

Normalización - Pauta Nro. 3 - NULLs

● Pauta Nro. 3.

- Evitar asignar atributos a relaciones, cuando estos frecuentemente pueden ser NULLs
- Si NULLs son inevitables, asegurar que las situaciones son excepcionales y no aplican a la mayoría de las tuplas

Normalización - Pauta Nro. 4 - Tuplas Espúreas

● Ejemplo. Esquema original

EMPLEADO_PROYECTO

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_PROYECTO	P_Ubicación
Diego	20222333	11/12/1970	5	Argentina
Laura	33456234	02/04/1985	5	Argentina
Marina	45432345	23/07/2006	2	Uruguay
Santiago	24345345	18/02/1975	5	Argentina

● Descomposición.

E_DNI	E.Fecha_Nacimiento	Nro_PROYECTO	P_Ubicación
20222333	11/12/1970	5	Argentina
33456234	02/04/1985	5	Argentina
45432345	23/07/2006	2	Uruguay
24345345	18/02/1975	5	Argentina

E_Nombre	P_Ubicación
Diego	Argentina
Laura	Argentina
Marina	Uruguay
Santiago	Argentina

Normalización - Pauta Nro. 4 - Tuplas Espúreas

● Ejemplo. Esquema original

EMPLEADO_PROYECTO

<u>E_Nombre</u>	<u>E_DNI</u>	<u>E.Fecha_Nacimiento</u>	<u>Nro_PROYECTO</u>	<u>P_Ubicación</u>
Diego	20222333	11/12/1970	5	Argentina
Laura	33456234	02/04/1985	5	Argentina
Marina	45432345	23/07/2006	2	Uruguay
Santiago	24345345	18/02/1975	5	Argentina

● Descomposición.

<u>E_DNI</u>	<u>E.Fecha_Nacimiento</u>	<u>Nro_PROYECTO</u>	<u>P_Ubicación</u>
20222333	11/12/1970	5	Argentina
33456234	02/04/1985	5	Argentina
45432345	23/07/2006	2	Uruguay
24345345	18/02/1975	5	Argentina

<u>E_Nombre</u>	<u>P_Ubicación</u>
Diego	Argentina
Laura	Argentina
Marina	Uruguay
Santiago	Argentina

● ¿Qué problema genera esta descomposición?

Normalización - Pauta Nro. 4 - Tuplas Espúreas

● Ejemplo. Esquema original

EMPLEADO_PROYECTO

<u>E_Nombre</u>	<u>E_DNI</u>	<u>E.Fecha_Nacimiento</u>	<u>Nro_PROYECTO</u>	<u>P_Ubicación</u>
Diego	20222333	11/12/1970	5	Argentina
Laura	33456234	02/04/1985	5	Argentina
Marina	45432345	23/07/2006	2	Uruguay
Santiago	24345345	18/02/1975	5	Argentina

● Descomposición.

<u>E_DNI</u>	<u>E.Fecha_Nacimiento</u>	<u>Nro_PROYECTO</u>	<u>P_Ubicación</u>
20222333	11/12/1970	5	Argentina
33456234	02/04/1985	5	Argentina
45432345	23/07/2006	2	Uruguay
24345345	18/02/1975	5	Argentina

<u>E_Nombre</u>	<u>P_Ubicación</u>
Diego	Argentina
Laura	Argentina
Marina	Uruguay
Santiago	Argentina

● ¿Qué problema genera esta descomposición?

No permite recuperar información original de EMPLEADO_PROYECTO

Normalización - Pauta Nro. 4 - Tuplas Espúreas

● Ejemplo. Esquema original

EMPLEADO_PROYECTO

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_PROYECTO	P_Ubicación
Diego	20222333	11/12/1970	5	Argentina
Laura	33456234	02/04/1985	5	Argentina
Marina	45432345	23/07/2006	2	Uruguay
Santiago	24345345	18/02/1975	5	Argentina

● Descomposición.

E_DNI	E.Fecha_Nacimiento	Nro_PROYECTO	P_Ubicación
20222333	11/12/1970	5	Argentina
33456234	02/04/1985	5	Argentina
45432345	23/07/2006	2	Uruguay
24345345	18/02/1975	5	Argentina

E_Nombre	P_Ubicación
Diego	Argentina
Laura	Argentina
Marina	Uruguay
Santiago	Argentina

- ¿Qué problema genera esta descomposición?
No permite recuperar información original de EMPLEADO_PROYECTO
- ¿Cuál es el resultado de aplicar NATURAL JOIN?

Normalización - Pauta Nro. 4 - Tuplas Espúreas

● Ejemplo. Esquema original

EMPLEADO_PROYECTO

E_Nombre	E_DNI	E.Fecha_Nacimiento	Nro_PROYECTO	P_Ubicación
Diego	20222333	11/12/1970	5	Argentina
Laura	33456234	02/04/1985	5	Argentina
Marina	45432345	23/07/2006	2	Uruguay
Santiago	24345345	18/02/1975	5	Argentina

● Descomposición.

E_DNI	E.Fecha_Nacimiento	Nro_PROYECTO	P_Ubicación
20222333	11/12/1970	5	Argentina
33456234	02/04/1985	5	Argentina
45432345	23/07/2006	2	Uruguay
24345345	18/02/1975	5	Argentina

E_Nombre	P_Ubicación
Diego	Argentina
Laura	Argentina
Marina	Uruguay
Santiago	Argentina

● ¿Qué problema genera esta descomposición?

No permite recuperar información original de EMPLEADO_PROYECTO

● ¿Cuál es el resultado de aplicar NATURAL JOIN?

Produce **tuplas espúreas** (información no válida)

E_DNI	E.Fecha_Nacimiento	Nro_PROYECTO	P_Ubicación	E_Nombre
20222333	11/12/1970	5	Argentina	Diego
33456234	02/04/1985	5	Argentina	Diego
24345345	18/02/1975	5	Argentina	Diego

Normalización - Pauta Nro. 4 - Tuplas Espúreas

- **No deseable.** Esta descomposición no es deseable porque cuando se intenta la reconstrucción a través de NATURAL JOIN no se obtiene información correcta
- **Causa.** P_Ubicación, relaciona ambos esquemas, pero no es ni clave primaria ni clave foránea de ninguno de ellos

Normalización - Pauta Nro. 4 - Tuplas Espúreas

- **No deseable.** Esta descomposición no es deseable porque cuando se intenta la reconstrucción a través de NATURAL JOIN no se obtiene información correcta
- **Causa.** P_Ubicación, relaciona ambos esquemas, pero no es ni clave primaria ni clave foránea de ninguno de ellos
- **Pauta Nro. 4.**
 - Diseñar esquemas tal que puedan ser relacionados por atributos que se encuentren apropiadamente relacionados por medio de condiciones de igualdad entre ellos (clave primaria, clave foránea), para evitar generación de tuplas espúreas
 - Evitar relaciones que contengan atributos de matching que no sean combinación de claves foránea/clave primaria porque JOINS sobre ellos pueden producir tuplas espúreas

Dependencias Funcionales

- **Propósito.** Herramienta formal para el análisis de esquemas. Permite detectar y describir problemas descritos previamente
- **Informalmente.** Restricción entre dos conjuntos de atributos X e Y de una BD. Los valores que toman los atributos de Y dependen de los valores que tomen X

Dependencias Funcionales

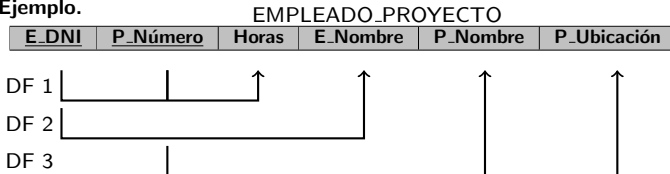
- **Propósito.** Herramienta formal para el análisis de esquemas. Permite detectar y describir problemas descriptos previamente
- **Informalmente.** Restricción entre dos conjuntos de atributos X e Y de una BD. Los valores que toman los atributos de Y dependen de los valores que tomen X
- **Ejemplo.**

EMPLEADO_PROYECTO

<u>E_DNI</u>	<u>P_Número</u>	Horas	E_Nombre	P_Nombre	P_Ubicación
--------------	-----------------	-------	----------	----------	-------------

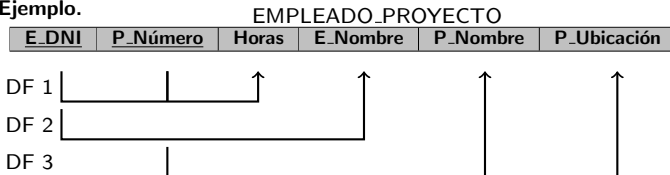
Dependencias Funcionales

- **Propósito.** Herramienta formal para el análisis de esquemas. Permite detectar y describir problemas descritos previamente
- **Informalmente.** Restricción entre dos conjuntos de atributos X e Y de una BD. Los valores que toman los atributos de Y dependen de los valores que tomen X
- **Ejemplo.**



Dependencias Funcionales

- **Propósito.** Herramienta formal para el análisis de esquemas. Permite detectar y describir problemas descritos previamente
- **Informalmente.** Restricción entre dos conjuntos de atributos X e Y de una BD. Los valores que toman los atributos de Y dependen de los valores que tomen X
- **Ejemplo.**



- **DFs.**
 - $\{E_DNI, P_Número\} \rightarrow Horas$
 - $E_DNI \rightarrow E_Nombre$
 - $P_Número \rightarrow \{P_Nombre, P_Ubicación\}$

Dependencias Funcionales

- **Formalmente.**

- Esquema relacional de la BD posee n atributos A_1, A_2, \dots, A_n
- Pensar toda la BD descrita por un solo esquema universal $R = \{A_1, A_2, \dots, A_n\}$. Esto no implica que realmente la BD se almacene como una tabla universal. Sólo se usará este concepto para construir la teoría formal de las dependencias de datos.
- **Definición.**
 - Sean X e Y dos conjuntos de atributos incluidos en R
 - La dependencia funcional (DF) indicada como $X \rightarrow Y$ especifica una restricción sobre las posibles tuplas que pueden conformar una instancia r de R
 - Restricción: para cualquiera dos tuplas t_1 y t_2 en r tal que $t_1[X] = t_2[X]$, se debe cumplir $t_1[Y] = t_2[Y]$

Dependencias Funcionales

- **Formalmente.**

- Esquema relacional de la BD posee n atributos A_1, A_2, \dots, A_n
- Pensar toda la BD descrita por un solo esquema universal $R = \{A_1, A_2, \dots, A_n\}$. Esto no implica que realmente la BD se almacene como una tabla universal. Sólo se usará este concepto para construir la teoría formal de las dependencias de datos.

- **Definición.**

- Sean X e Y dos conjuntos de atributos incluidos en R
- La dependencia funcional (DF) indicada como $X \rightarrow Y$ especifica una restricción sobre las posibles tuplas que pueden conformar una instancia r de R
- Restricción: para cualquiera dos tuplas t_1 y t_2 en r tal que $t_1[X] = t_2[X]$, se debe cumplir $t_1[Y] = t_2[Y]$

- **Ejemplo.**

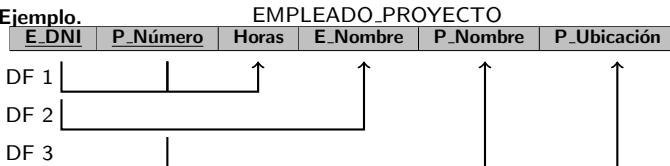
EMPLEADO_PROYECTO

E_DNI	P_Número	Horas	E_Nombre	P_Nombre	P_Ubicación
20222333	2	123	Diego	Área 51	C.A.B.A
20222333	5	12	Diego	Vaca viva	Neuquén

$E_DNI \rightarrow E_Nombre$

Dependencias Funcionales

- **Ejemplo.**

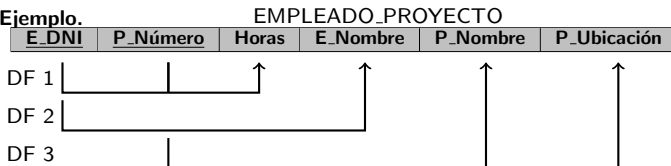


- **DFs.**

- $\{E_DNI, P_Número\} \rightarrow Horas$
- $E_DNI \rightarrow E_Nombre$
- $P_Número \rightarrow \{P_Nombre, P_Ubicación\}$

Dependencias Funcionales

● Ejemplo.



● DFs.

- {E_DNI, P_Número} → Horas
- E_DNI → E_Nombre
- P_Número → {P_Nombre, P_Ubicación}

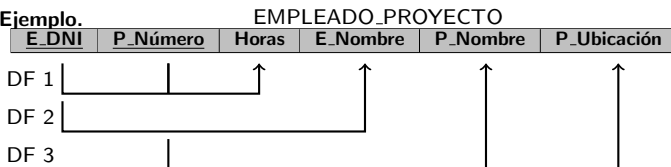
● Frase. "Y es funcionalmente dependiente de X"

● **Definición 1.** Conjunto de atributos *X* se denominan **lado izquierdo** de la DF

● **Definición 2.** Conjunto de atributos *Y* se denominan **lado derecho** de la DF

Dependencias Funcionales

● Ejemplo.



● DFs.

- {E_DNI, P_Número} → Horas
- E_DNI → E_Nombre
- P_Número → {P_Nombre, P_Ubicación}

● Frase. "Y es funcionalmente dependiente de X"

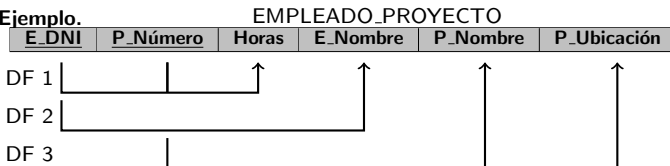
● Definición 1. Conjunto de atributos X se denominan **lado izquierdo** de la DF

● Definición 2. Conjunto de atributos Y se denominan **lado derecho** de la DF

Decidir si las siguientes propiedades son Verdaderas o Falsas

Dependencias Funcionales

● Ejemplo.



● DFs.

- $\{E_DNI, P_Número\} \rightarrow Horas$
- $E_DNI \rightarrow E_Nombre$
- $P_Número \rightarrow \{P_Nombre, P_Ubicación\}$

● Frase. "Y es funcionalmente dependiente de X"

● Definición 1. Conjunto de atributos X se denominan **lado izquierdo** de la DF

● Definición 2. Conjunto de atributos Y se denominan **lado derecho** de la DF

Decidir si las siguientes propiedades son Verdaderas o Falsas

- **Propiedad 1.** Si X es clave candidata (CK) de R, entonces $X \rightarrow Y \forall$ subconjunto de atributos Y de R.
- **Propiedad 1'.** Si X es CK de R, entonces $X \rightarrow R$.
- **Propiedad 2.** $X \rightarrow Y$ implica $Y \rightarrow X$.

Dependencias Funcionales

- **Semántica.** DF son propiedad de la semántica (o significado) de los atributos.
- **Diseño.** Diseñadores de las BD deben usar su entendimiento de la semántica de atributos de R para especificar las DF y deberán respetar TODOS los $r(R)$

Dependencias Funcionales

- **Semántica.** DF son propiedad de la semántica (o significado) de los atributos.
- **Diseño.** Diseñadores de las BD deben usar su entendimiento de la semántica de atributos de R para especificar las DF y deberán respetar TODOS los $r(R)$
- **Instancias legales.** $r(R)$ que satisface restricciones de DF se denomina instancia legal, estado legal o extensión legal de R
- **Inferencia de DF.** Dada una relación con sus datos, no es posible determinar sus DF a través de sus valores. Es necesario conocer el significado y relación que existe entre los atributos que la componen

Dependencias Funcionales

- **Semántica.** DF son propiedad de la semántica (o significado) de los atributos.
- **Diseño.** Diseñadores de las BD deben usar su entendimiento de la semántica de atributos de R para especificar las DF y deberán respetar TODOS los $r(R)$
- **Instancias legales.** $r(R)$ que satisface restricciones de DF se denomina instancia legal, estado legal o extensión legal de R
- **Inferencia de DF.** Dada una relación con sus datos, no es posible determinar sus DF a través de sus valores. Es necesario conocer el significado y relación que existe entre los atributos que la componen
- **Ejemplo.** ¿Cuáles son las DF? DICTA

Profesor	Curso	Libro
Pérez	Algo I	Dijkstra
Fernández	Algo II	Ullman
Ruz	BDs	Elmasri
Pérez	SO	Tanenbaum

Dependencias Funcionales

- **Semántica.** DF son propiedad de la semántica (o significado) de los atributos.
- **Diseño.** Diseñadores de las BD deben usar su entendimiento de la semántica de atributos de R para especificar las DF y deberán respetar TODOS los $r(R)$
- **Instancias legales.** $r(R)$ que satisface restricciones de DF se denomina instancia legal, estado legal o extensión legal de R
- **Inferencia de DF.** Dada una relación con sus datos, no es posible determinar sus DF a través de sus valores. Es necesario conocer el significado y relación que existe entre los atributos que la componen
- **Ejemplo. ¿Cuáles son las DF?** DICTA

Profesor	Curso	Libro
Pérez	Algo I	Dijkstra
Fernández	Algo II	Ullman
Ruz	BDs	Elmasri
Pérez	SO	Tanenbaum

- **Existencia.** Una DF puede existir si la cumple una instancia $r(R)$
 - Para “confirmar” la existencia de una DF es necesario conocer la semántica de sus atributos
 - Para “descartar” la existencia de una DF sólo basta mostrar la existencia de tuplas que violan dicha “potencial” DF

Dependencias Funcionales

- **Semántica.** DF son propiedad de la semántica (o significado) de los atributos.
- **Diseño.** Diseñadores de las BD deben usar su entendimiento de la semántica de atributos de R para especificar las DF y deberán respetar TODOS los $r(R)$
- **Instancias legales.** $r(R)$ que satisface restricciones de DF se denomina instancia legal, estado legal o extensión legal de R
- **Inferencia de DF.** Dada una relación con sus datos, no es posible determinar sus DF a través de sus valores. Es necesario conocer el significado y relación que existe entre los atributos que la componen
- **Ejemplo.** ¿Cuáles son las DF? DICTA

Profesor	Curso	Libro
Pérez	Algo I	Dijkstra
Fernández	Algo II	Ullman
Ruz	BDs	Elmasri
Pérez	SO	Tanenbaum

- **Existencia.** Una DF puede existir si la cumple una instancia $r(R)$
 - Para “confirmar” la existencia de una DF es necesario conocer la semántica de sus atributos
 - Para “descartar” la existencia de una DF sólo basta mostrar la existencia de tuplas que violan dicha “potencial” DF
- **Notación.** Conjunto de DF, se denota como F
- **Inferencia.** Diseñador especifica DFs que son semánticamente obvias. Existen

FN basadas en PK

- **Se asume.**
 - Se cuenta con el conjunto de DF para cada relación
 - Cada relación tiene designada su Clave Primaria (PK)

FN basadas en PK

- Se asume.
 - Se cuenta con el conjunto de DF para cada relación
 - Cada relación tiene designada su Clave Primaria (PK)
- Proceso de Normalización.
 - Propuesto por Codd (1972a)
 - A cada esquema ejecutarle una serie de test para **certificar** que satisface una **forma normal**

FN basadas en PK

- Se asume.
 - Se cuenta con el conjunto de DF para cada relación
 - Cada relación tiene designada su Clave Primaria (PK)
- Proceso de Normalización.
 - Propuesto por Codd (1972a)
 - A cada esquema ejecutarle una serie de test para **certificar** que satisface una **forma normal**
- Normalización de los datos.
 - Proceso de analizar los esquemas, basándose en DF y PK
 - Objetivo: lograr propiedades deseables
 - Minimizar redundancia
 - Minimizar anomalías de inserción, deletión y modificación
 - Esquemas que no pasan ciertos test de **formas normales**, se decomponen en esquemas más pequeños que pasan el test (y sus propiedades)

FN basadas en PK

- **Se asume.**
 - Se cuenta con el conjunto de DF para cada relación
 - Cada relación tiene designada su Clave Primaria (PK)
- **Proceso de Normalización.**
 - Propuesto por Codd (1972a)
 - A cada esquema ejecutarle una serie de test para **certificar** que satisface una **forma normal**
- **Normalización de los datos.**
 - Proceso de analizar los esquemas, basándose en DF y PK
 - Objetivo: lograr propiedades deseables
 - Minimizar redundancia
 - Minimizar anomalías de inserción, deletión y modificación
 - Esquemas que no pasan ciertos test de **formas normales**, se decomponen en esquemas más pequeños que pasan el test (y sus propiedades)
- **Definición.** La **forma normal** de una relación refiere a la mayor forma normal alcanzada por ella

FN basadas en PK

- **Sin garantía.** Las formas normales, consideradas aisladas de otros factores, no garantizan un buen diseño de la BD

FN basadas en PK

- **Sin garantía.** Las formas normales, consideradas aisladas de otros factores, no garantizan un buen diseño de la BD
- **Propiedades.** Luego de proceso de normalización por descomposición
 - **Nonadditive Join (Lossless Join).** Garantía de que no ocurre problema de generación de tuplas espúreas. La relación original tiene que poder ser recuperada de la descomposición.
 - **Preservación de DF.** Garantía de que cada DF se encuentra representada en algún esquema resultante de la descomposición
- **Lossless Join** debe lograrse a cualquier costo
- **Preservación de DF.** Es deseable, pero en algunos casos es sacrificada

FN basadas en PK

- Súper Clave (SK).

FN basadas en PK

- **Súper Clave (SK).** Una SK de $R = \{A_1, A_2, \dots, A_n\}$ es un subconjunto de atributos $S \subseteq R$ con la propiedad de que no hay dos tuplas t_1, t_2 en un estado legal $r(R)$ que cumplan $t_1(S) = t_2(S)$

FN basadas en PK

- **Súper Clave (SK).** Una SK de $R = \{A_1, A_2, \dots, A_n\}$ es un subconjunto de atributos $S \subseteq R$ con la propiedad de que no hay dos tuplas t_1, t_2 en un estado legal $r(R)$ que cumplan $t_1(S) = t_2(S)$
- **Clave (K).**

FN basadas en PK

- **Súper Clave (SK).** Una SK de $R = \{A_1, A_2, \dots, A_n\}$ es un subconjunto de atributos $S \subseteq R$ con la propiedad de que no hay dos tuplas t_1, t_2 en un estado legal $r(R)$ que cumplan $t_1(S) = t_2(S)$
- **Clave (K).** Una clave K es una SK con la propiedad adicional de que al remover cualquier atributo de K , deja de ser SK. Es decir, K es una SK *minimal*

FN basadas en PK

- **Súper Clave (SK).** Una SK de $R = \{A_1, A_2, \dots, A_n\}$ es un subconjunto de atributos $S \subseteq R$ con la propiedad de que no hay dos tuplas t_1, t_2 en un estado legal $r(R)$ que cumplan $t_1(S) = t_2(S)$
- **Clave (K).** Una clave K es una SK con la propiedad adicional de que al remover cualquier atributo de K , deja de ser SK. Es decir, K es una SK *minimal*
- **Clave Candidata (CK).**

FN basadas en PK

- **Súper Clave (SK).** Una SK de $R = \{A_1, A_2, \dots, A_n\}$ es un subconjunto de atributos $S \subseteq R$ con la propiedad de que no hay dos tuplas t_1, t_2 en un estado legal $r(R)$ que cumplan $t_1(S) = t_2(S)$
- **Clave (K).** Una clave K es una SK con la propiedad adicional de que al remover cualquier atributo de K , deja de ser SK. Es decir, K es una SK *minimal*
- **Clave Candidata (CK).** Si un esquema posee más de una clave, cada una de ellas se denominan clave candidata

FN basadas en PK

- **Súper Clave (SK).** Una SK de $R = \{A_1, A_2, \dots, A_n\}$ es un subconjunto de atributos $S \subseteq R$ con la propiedad de que no hay dos tuplas t_1, t_2 en un estado legal $r(R)$ que cumplan $t_1(S) = t_2(S)$
- **Clave (K).** Una clave K es una SK con la propiedad adicional de que al remover cualquier atributo de K , deja de ser SK. Es decir, K es una SK *minimal*
- **Clave Candidata (CK).** Si un esquema posee más de una clave, cada una de ellas se denominan clave candidata
- **Clave Primaria (PK).**

FN basadas en PK

- **Súper Clave (SK).** Una SK de $R = \{A_1, A_2, \dots, A_n\}$ es un subconjunto de atributos $S \subseteq R$ con la propiedad de que no hay dos tuplas t_1, t_2 en un estado legal $r(R)$ que cumplan $t_1(S) = t_2(S)$
- **Clave (K).** Una clave K es una SK con la propiedad adicional de que al remover cualquier atributo de K , deja de ser SK. Es decir, K es una SK *minimal*
- **Clave Candidata (CK).** Si un esquema posee más de una clave, cada una de ellas se denominan clave candidata
- **Clave Primaria (PK).** Una de las CK es designada *arbitrariamente* como PK

FN basadas en PK

- **Súper Clave (SK).** Una SK de $R = \{A_1, A_2, \dots, A_n\}$ es un subconjunto de atributos $S \subseteq R$ con la propiedad de que no hay dos tuplas t_1, t_2 en un estado legal $r(R)$ que cumplan $t_1(S) = t_2(S)$
- **Clave (K).** Una clave K es una SK con la propiedad adicional de que al remover cualquier atributo de K , deja de ser SK. Es decir, K es una SK *minimal*
- **Clave Candidata (CK).** Si un esquema posee más de una clave, cada una de ellas se denominan clave candidata
- **Clave Primaria (PK).** Una de las CK es designada *arbitrariamente* como PK
- **Clave Secundaria.**

FN basadas en PK

- **Súper Clave (SK).** Una SK de $R = \{A_1, A_2, \dots, A_n\}$ es un subconjunto de atributos $S \subseteq R$ con la propiedad de que no hay dos tuplas t_1, t_2 en un estado legal $r(R)$ que cumplan $t_1(S) = t_2(S)$
- **Clave (K).** Una clave K es una SK con la propiedad adicional de que al remover cualquier atributo de K , deja de ser SK. Es decir, K es una SK *minimal*
- **Clave Candidata (CK).** Si un esquema posee más de una clave, cada una de ellas se denominan clave candidata
- **Clave Primaria (PK).** Una de las CK es designada *arbitrariamente* como PK
- **Clave Secundaria.** CK que no es PK

FN basadas en PK

- **Súper Clave (SK).** Una SK de $R = \{A_1, A_2, \dots, A_n\}$ es un subconjunto de atributos $S \subseteq R$ con la propiedad de que no hay dos tuplas t_1, t_2 en un estado legal $r(R)$ que cumplan $t_1(S) = t_2(S)$
- **Clave (K).** Una clave K es una SK con la propiedad adicional de que al remover cualquier atributo de K , deja de ser SK. Es decir, K es una SK *minimal*
- **Clave Candidata (CK).** Si un esquema posee más de una clave, cada una de ellas se denominan clave candidata
- **Clave Primaria (PK).** Una de las CK es designada *arbitrariamente* como PK
- **Clave Secundaria.** CK que no es PK
- **Atributo primo.**

FN basadas en PK

- **Súper Clave (SK).** Una SK de $R = \{A_1, A_2, \dots, A_n\}$ es un subconjunto de atributos $S \subseteq R$ con la propiedad de que no hay dos tuplas t_1, t_2 en un estado legal $r(R)$ que cumplan $t_1(S) = t_2(S)$
- **Clave (K).** Una clave K es una SK con la propiedad adicional de que al remover cualquier atributo de K , deja de ser SK. Es decir, K es una SK *minimal*
- **Clave Candidata (CK).** Si un esquema posee más de una clave, cada una de ellas se denominan clave candidata
- **Clave Primaria (PK).** Una de las CK es designada *arbitrariamente* como PK
- **Clave Secundaria.** CK que no es PK
- **Atributo primo.** Atributo de un esquema R que pertenece a alguna CK de R

FN basadas en PK

- **Súper Clave (SK).** Una SK de $R = \{A_1, A_2, \dots, A_n\}$ es un subconjunto de atributos $S \subseteq R$ con la propiedad de que no hay dos tuplas t_1, t_2 en un estado legal $r(R)$ que cumplan $t_1(S) = t_2(S)$
- **Clave (K).** Una clave K es una SK con la propiedad adicional de que al remover cualquier atributo de K , deja de ser SK. Es decir, K es una SK *minimal*
- **Clave Candidata (CK).** Si un esquema posee más de una clave, cada una de ellas se denominan clave candidata
- **Clave Primaria (PK).** Una de las CK es designada *arbitrariamente* como PK
- **Clave Secundaria.** CK que no es PK
- **Atributo primo.** Atributo de un esquema R que pertenece a alguna CK de R
- **Requisito.** En la práctica, todos los esquemas deben poseer PK

FN basadas en PK

- **Súper Clave (SK).** Una SK de $R = \{A_1, A_2, \dots, A_n\}$ es un subconjunto de atributos $S \subseteq R$ con la propiedad de que no hay dos tuplas t_1, t_2 en un estado legal $r(R)$ que cumplan $t_1(S) = t_2(S)$
- **Clave (K).** Una clave K es una SK con la propiedad adicional de que al remover cualquier atributo de K , deja de ser SK. Es decir, K es una SK *minimal*
- **Clave Candidata (CK).** Si un esquema posee más de una clave, cada una de ellas se denominan clave candidata
- **Clave Primaria (PK).** Una de las CK es designada *arbitrariamente* como PK
- **Clave Secundaria.** CK que no es PK
- **Atributo primo.** Atributo de un esquema R que pertenece a alguna CK de R
- **Requisito.** En la práctica, todos los esquemas deben poseer PK
- **Ejercicio.** Proponer un Esquema con todos estos elementos e identificarlos

FN basadas en PK - 1FN

- **1FN.**

- **Prohíbe** relaciones dentro de relaciones o relaciones como valores de atributos dentro de tuplas
- **Admite** El dominio de un atributo debe incluir sólo valores atómicos (simples e indivisibles). En la tupla, puede tomar 1 solo valor del dominio.

FN basadas en PK - 1FN

- **1FN.**
 - **Prohíbe** relaciones dentro de relaciones o relaciones como valores de atributos dentro de tuplas
 - **Admite** El dominio de un atributo debe incluir sólo valores atómicos (simples e indivisibles). En la tupla, puede tomar 1 solo valor del dominio.
- **Ejemplo.**

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Areas_Influencia
Investigación	2	27-23345876-9	{Argentina, Brasil, Uruguay}
Prensa	3	20-17283948-4	{Chile}
Administración	8	27-38476827-2	{Argentina}

- ¿Está en 1FN?

FN basadas en PK - 1FN

- **1FN.**
 - **Prohíbe** relaciones dentro de relaciones o relaciones como valores de atributos dentro de tuplas
 - **Admite** El dominio de un atributo debe incluir sólo valores atómicos (simples e indivisibles). En la tupla, puede tomar 1 solo valor del dominio.
- **Ejemplo.**

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Areas_Influencia
Investigación	2	27-23345876-9	{Argentina, Brasil, Uruguay}
Prensa	3	20-17283948-4	{Chile}
Administración	8	27-38476827-2	{Argentina}

- ¿Está en 1FN? ¡No! D_Areas_Influencia no es un atributo atómico

FN basadas en PK - 1FN

- **Técnicas para alcanzar 1FN.**

- ➊ Remover atributo que viola 1FN y ubicarlo en una nueva relación, DEPTO_AREAS, junto con la PK D_Número. La nueva relación tiene como PK ambos atributos

FN basadas en PK - 1FN

- **Técnicas para alcanzar 1FN.**

- ➊ Remover atributo que viola 1FN y ubicarlo en una nueva relación, DEPTO_AREAS, junto con la PK D_Número. La nueva relación tiene como PK ambos atributos

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Areas_Influencia
Investigación	2	27-23345876-9	{Argentina, Brasil, Uruguay}
Prensa	3	20-17283948-4	{Chile}
Administración	8	27-38476827-2	{Argentina}

FN basadas en PK - 1FN

- **Técnicas para alcanzar 1FN.**

- 1 Remover atributo que viola 1FN y ubicarlo en una nueva relación, DEPTO_AREAS, junto con la PK D_Número. La nueva relación tiene como PK ambos atributos

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Areas_Influencia
Investigación	2	27-23345876-9	{Argentina, Brasil, Uruguay}
Prensa	3	20-17283948-4	{Chile}
Administración	8	27-38476827-2	{Argentina}

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL
Investigación	2	27-23345876-9
Prensa	3	20-17283948-4
Administración	8	27-38476827-2

DEPTO_AREAS

D_Número	D_Areas_Influencia
2	Argentina
2	Brasil
2	Uruguay
3	Chile
8	Argentina

FN basadas en PK - 1FN

- **Técnicas para alcanzar 1FN.**
 - ② Expandir la PK que permita que exista más de un mismo D_Número, pero con distinta área de influencia.

FN basadas en PK - 1FN

- Técnicas para alcanzar 1FN.
 - ② Expandir la PK que permita que exista más de un mismo D_Número, pero con distinta área de influencia.

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Areas_Influencia
Investigación	2	27-23345876-9	{Argentina, Brasil, Uruguay}
Prensa	3	20-17283948-4	{Chile}
Administración	8	27-38476827-2	{Argentina}

FN basadas en PK - 1FN

- Técnicas para alcanzar 1FN.
 - 2 Expandir la PK que permita que exista más de un mismo D_Número, pero con distinta área de influencia.

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Areas_Influencia
Investigación	2	27-23345876-9	{Argentina, Brasil, Uruguay}
Prensa	3	20-17283948-4	{Chile}
Administración	8	27-38476827-2	{Argentina}

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Area_Influencia
Investigación	2	27-23345876-9	Argentina
Investigación	2	27-23345876-9	Brasil
Investigación	2	27-23345876-9	Uruguay
Prensa	3	20-17283948-4	Chile
Administración	8	27-38476827-2	Argentina

- ¿Qué problema tiene esta solución?

FN basadas en PK - 1FN

- Técnicas para alcanzar 1FN.
 - 2 Expandir la PK que permita que exista más de un mismo D_Número, pero con distinta área de influencia.

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Areas_Influencia
Investigación	2	27-23345876-9	{Argentina, Brasil, Uruguay}
Prensa	3	20-17283948-4	{Chile}
Administración	8	27-38476827-2	{Argentina}

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Area_Influencia
Investigación	2	27-23345876-9	Argentina
Investigación	2	27-23345876-9	Brasil
Investigación	2	27-23345876-9	Uruguay
Prensa	3	20-17283948-4	Chile
Administración	8	27-38476827-2	Argentina

- ¿Qué problema tiene esta solución? Introduce *redundancia* en la relación

FN basadas en PK - 1FN

- **Técnicas para alcanzar 1FN.**

- ④ Si se conoce la máxima cantidad de valores que puede tomar el atributo, se pueden generar tantos atributos como esa cantidad.

FN basadas en PK - 1FN

- **Técnicas para alcanzar 1FN.**

- ③ Si se conoce la máxima cantidad de valores que puede tomar el atributo, se pueden generar tantos atributos como esa cantidad.

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Areas_Influencia
Investigación	2	27-23345876-9	{Argentina, Brasil, Uruguay}
Prensa	3	20-17283948-4	{Chile}
Administración	8	27-38476827-2	{Argentina}

FN basadas en PK - 1FN

- **Técnicas para alcanzar 1FN.**

- ③ Si se conoce la máxima cantidad de valores que puede tomar el atributo, se pueden generar tantos atributos como esa cantidad.

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Areas_Influencia
Investigación	2	27-23345876-9	{Argentina, Brasil, Uruguay}
Prensa	3	20-17283948-4	{Chile}
Administración	8	27-38476827-2	{Argentina}

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Area_Influencia_1	D_Area_Influencia_2	D_Area_Influencia_3
Investigación	2	27-23345876-9	Uruguay	Brasil	Argentina
Prensa	3	20-17283948-4	Chile	NULL	NULL
Administración	8	27-38476827-2	Argentina	NULL	NULL

FN basadas en PK - 1FN

- Técnicas para alcanzar 1FN.

- ③ Si se conoce la máxima cantidad de valores que puede tomar el atributo, se pueden generar tantos atributos como esa cantidad.

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Areas_Influencia
Investigación	2	27-23345876-9	{Argentina, Brasil, Uruguay}
Prensa	3	20-17283948-4	{Chile}
Administración	8	27-38476827-2	{Argentina}

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Area_Influencia_1	D_Area_Influencia_2	D_Area_Influencia_3
Investigación	2	27-23345876-9	Uruguay	Brasil	Argentina
Prensa	3	20-17283948-4	Chile	NULL	NULL
Administración	8	27-38476827-2	Argentina	NULL	NULL

- ¿Qué problema tiene esta solución?

FN basadas en PK - 1FN

- **Técnicas para alcanzar 1FN.**

- ③ Si se conoce la máxima cantidad de valores que puede tomar el atributo, se pueden generar tantos atributos como esa cantidad.

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Areas_Influencia
Investigación	2	27-23345876-9	{Argentina, Brasil, Uruguay}
Prensa	3	20-17283948-4	{Chile}
Administración	8	27-38476827-2	{Argentina}

DEPARTAMENTO

D_NOMBRE	D_Número	D_MGR_CUIL	D_Area_Influencia_1	D_Area_Influencia_2	D_Area_Influencia_3
Investigación	2	27-23345876-9	Uruguay	Brasil	Argentina
Prensa	3	20-17283948-4	Chile	NULL	NULL
Administración	8	27-38476827-2	Argentina	NULL	NULL

- ¿Qué problema tiene esta solución?
 - Introducción de valores NULL en casos que la tupla no posee 3 valores para área
 - ¿Cuál es la semántica en cuanto a la ubicación de los valores de área?
 - Consultas acerca del área se vuelven más complejas. Ej. Listar todos los Departamentos cuya área de influencia incluye a “Argentina”

FN basadas en PK - 1FN

- **Mejor solución.** La primer opción suele ser la mejor porque no sufre de redundancia y es genérica (no se limita a un máximo de valores posibles)
- **Recursividad.** La Técnica se puede utilizar recursivamente para múltiples niveles

FN basadas en PK - 1FN

- **Mejor solución.** La primer opción suele ser la mejor porque no sufre de redundancia y es genérica (no se limita a un máximo de valores posibles)
- **Recursividad.** La Técnica se puede utilizar recursivamente para múltiples niveles
- **Múltiples atributos multivaludados.** Debe manejarse con cuidado
- **Ejemplo.**

PERSONA

P_CUIL	P_Cédula_Azul	Teléfonos
27-23345876-9	{ JYF 456, PFR 345, KOL 102 }	{ 11-4567-2321, 11-6783-9283 }
20-17283948-4	{ RUI 234, FGH 736 }	{ 2345-423-3456, 11-2343-2342, 11-2321-2321 }

FN basadas en PK - 1FN

- **Mejor solución.** La primer opción suele ser la mejor porque no sufre de redundancia y es genérica (no se limita a un máximo de valores posibles)
- **Recursividad.** La Técnica se puede utilizar recursivamente para múltiples niveles
- **Múltiples atributos multivaludados.** Debe manejarse con cuidado
- **Ejemplo.**

PERSONA

P_CUIL	P_Cédula_Azul	Teléfonos
27-23345876-9	{JYF 456, PFR 345, KOL 102}	{11-4567-2321, 11-6783-9283}
20-17283948-4	{RUI 234, FGH 736}	{2345-423-3456, 11-2343-2342, 11-2321-2321}

- **Aplicando “textualmente” Estrategia Nro. 2.**
PERSONA_CEDULA_TELÉFONO

P_CUIL	P_Cédula_Azul	P_Teléfono
--------	---------------	------------

- ¿Qué problema produce?

FN basadas en PK - 1FN

- **Mejor solución.** La primer opción suele ser la mejor porque no sufre de redundancia y es genérica (no se limita a un máximo de valores posibles)
- **Recursividad.** La Técnica se puede utilizar recursivamente para múltiples niveles
- **Múltiples atributos multivaludados.** Debe manejarse con cuidado
- **Ejemplo.**

PERSONA

P_CUIL	P_Cédula_Azul	Teléfonos
27-23345876-9	{JYF 456, PFR 345, KOL 102}	{11-4567-2321, 11-6783-9283}
20-17283948-4	{RUI 234, FGH 736}	{2345-423-3456, 11-2343-2342, 11-2321-2321}

- **Aplicando “textualmente” Estrategia Nro. 2.**
PERSONA_CEDULA_TELÉFONO

P_CUIL	P_Cédula_Azul	P_Teléfono
--------	---------------	------------

- ¿Qué problema produce? **Genera relación no existente entre P_Cédula_Azul y P_Teléfono**

FN basadas en PK - 1FN

- **Mejor solución.** La primer opción suele ser la mejor porque no sufre de redundancia y es genérica (no se limita a un máximo de valores posibles)
- **Recursividad.** La Técnica se puede utilizar recursivamente para múltiples niveles
- **Múltiples atributos multivaludados.** Debe manejarse con cuidado
- **Ejemplo.**

PERSONA

P_CUIL	P_Cédula_Azul	Teléfonos
27-23345876-9	{JYF 456, PFR 345, KOL 102}	{11-4567-2321, 11-6783-9283}
20-17283948-4	{RUI 234, FGH 736}	{2345-423-3456, 11-2343-2342, 11-2321-2321}

- **Aplicando “textualmente” Estrategia Nro. 2.**
PERSONA_CEDULA_TELÉFONO

P_CUIL	P_Cédula_Azul	P_Teléfono
--------	---------------	------------

- ¿Qué problema produce? **Genera relación no existente entre P_Cédula_Azul y P_Teléfono**

FN basadas en PK - 1FN

- **Mejor solución.** La primer opción suele ser la mejor porque no sufre de redundancia y es genérica (no se limita a un máximo de valores posibles)
- **Recursividad.** La Técnica se puede utilizar recursivamente para múltiples niveles
- **Múltiples atributos multivaludados.** Debe manejarse con cuidado
- **Ejemplo.**

PERSONA

P_CUIL	P_Cédula_Azul	Teléfonos
27-23345876-9	{JYF 456, PFR 345, KOL 102}	{11-4567-2321, 11-6783-9283}
20-17283948-4	{RUI 234, FGH 736}	{2345-423-3456, 11-2343-2342, 11-2321-2321}

- **Aplicando “textualmente” Estrategia Nro. 2.**

PERSONA_CEDULA_TELÉFONO

P_CUIL	P_Cédula_Azul	P_Teléfono
--------	---------------	------------

- ¿Qué problema produce? **Genera relación no existente entre P_Cédula_Azul y P_Teléfono**
- **Solución.**

FN basadas en PK - 1FN

- **Mejor solución.** La primer opción suele ser la mejor porque no sufre de redundancia y es genérica (no se limita a un máximo de valores posibles)
- **Recursividad.** La Técnica se puede utilizar recursivamente para múltiples niveles
- **Múltiples atributos multivaludados.** Debe manejarse con cuidado
- **Ejemplo.**

PERSONA

P_CUIL	P_Cédula_Azul	Teléfonos
27-23345876-9	{JYF 456, PFR 345, KOL 102}	{11-4567-2321, 11-6783-9283}
20-17283948-4	{RUI 234, FGH 736}	{2345-423-3456, 11-2343-2342, 11-2321-2321}

- **Aplicando “textualmente” Estrategia Nro. 2.**

PERSONA_CEDULA_TELÉFONO

P_CUIL	P_Cédula_Azul	P_Teléfono
--------	---------------	------------

- ¿Qué problema produce? **Genera relación no existente entre P_Cédula_Azul y P_Teléfono**
- **Solución.** Utilizar Estrategia Nro. 1

PERSONA_CÉDULA

P_CUIL	P_Cédula_Azul
--------	---------------

PERSONA_TELÉFONO

P_CUIL	P_Teléfonos
--------	-------------

FN basadas en PK - 1FN

- **Relaciones anidadas.** Cuando el valor de una tupla es una relación.
- 1NF prohíbe relaciones anidadas

FN basadas en PK - 1FN

● Ejemplo.

EMP_PROY

<u>E_CUIL</u>	E_Nombre	Proyectos	
		P_Número	Horas
27-23345876-9	Diego	1	20,5
		2	3,5
20-17283948-4	Laura	4	10
27-38476827-2	Marina	2	7,5
		4	11,5
		7	3,0
...

- E_CUIL es PK de EMP_PROY. P_Número es clave parcial de relación anidada

FN basadas en PK - 1FN

● Ejemplo.

EMP_PROY

E_CUIL	E_Nombre	Proyectos	
		P_Número	Horas
27-23345876-9	Diego	1	20,5
		2	3,5
20-17283948-4	Laura	4	10
27-38476827-2	Marina	2	7,5
		4	11,5
		7	3,0
...

- E_CUIL es PK de EMP_PROY. P_Número es clave parcial de relación anidada
- Técnica para alcanzar 1FN.
 - Mover atributos de relación anidada a una nueva relación
 - Agregar a la nueva relación la PK de relación original
 - PK de la nueva relación es: Clave parcial + PK relación original

FN basadas en PK - 1FN

● Ejemplo.

EMP_PROY

E_CUIL	E_Nombre	Proyectos	
		P_Número	Horas
27-23345876-9	Diego	1	20,5
		2	3,5
20-17283948-4	Laura	4	10
27-38476827-2	Marina	2	7,5
		4	11,5
		7	3,0
...

- E_CUIL es PK de EMP_PROY. P_Número es clave parcial de relación anidada
- Técnica para alcanzar 1FN.
 - Mover atributos de relación anidada a una nueva relación
 - Agregar a la nueva relación la PK de relación original
 - PK de la nueva relación es: Clave parcial + PK relación original

EMP

E_CUIL	E_Nombre
27-23345876-9	Diego
20-17283948-4	Laura
27-38476827-2	Marina

EMP_PROY

E_CUIL	P_Número	Horas
27-23345876-9	1	20,5
27-23345876-9	2	3,5
20-17283948-4	4	10
27-38476827-2	2	7,5
27-38476827-2	4	11,5
27-38476827-2	7	3,0

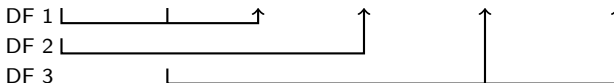
FN basadas en PK - 2FN

- **DF Completa.** Una DF $X \rightarrow Y$ es Completa si al eliminar algún atributo A de X la DF deja de existir
- **DF Parcial.** Una DF $X \rightarrow Y$ es Parcial si es posible eliminar algún atributo A de X y la DF continúa existiendo

FN basadas en PK - 2FN

- **DF Completa.** Una DF $X \rightarrow Y$ es Completa si al eliminar algún atributo A de X la DF deja de existir
- **DF Parcial.** Una DF $X \rightarrow Y$ es Parcial si es posible eliminar algún atributo A de X y la DF continúa existiendo
- **Ejemplo.**

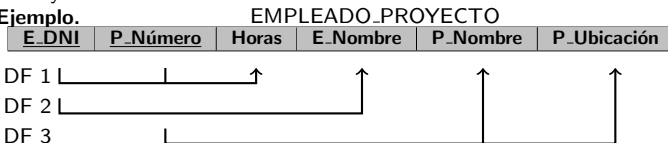
EMPLEADO_PROYECTO					
<u>E_DNI</u>	<u>P_Número</u>	Horas	E_Nombre	P_Nombre	P_Ubicación



- Horas depende de manera Completa de PK
- E_Nombre depende de manera Parcial de PK
- P_Nombre y P_Ubicación dependen de manera Parcial de PK

FN basadas en PK - 2FN

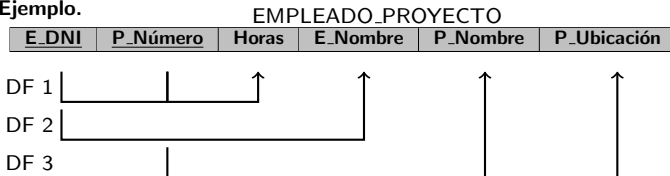
- **DF Completa.** Una DF $X \rightarrow Y$ es Completa si al eliminar algún atributo A de X la DF deja de existir
- **DF Parcial.** Una DF $X \rightarrow Y$ es Parcial si es posible eliminar algún atributo A de X y la DF continúa existiendo
- **Ejemplo.**



- Horas depende de manera Completa de PK
- E_Nombre depende de manera Parcial de PK
- P_Nombre y P_Ubicación dependen de manera Parcial de PK
- **2FN.** Un esquema R está en 2FN si todo atributo no primo A de R depende funcionalmente de manera completa de la PK de R

FN basadas en PK - 2FN

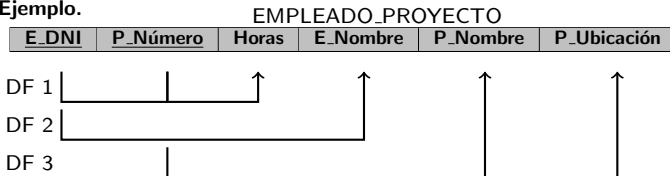
- Ejemplo.



- ¿Está en 2FN?

FN basadas en PK - 2FN

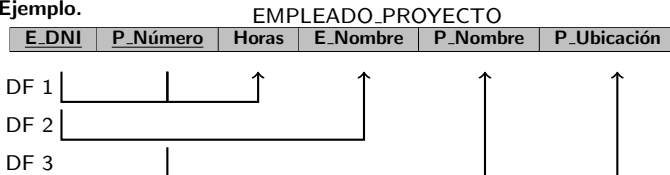
- Ejemplo.



- ¿Está en 2FN? ¡NO! Se ve, por DF 2 y DF 3, que hay atributos que dependen parcialmente de la PK

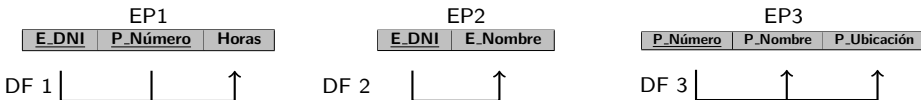
FN basadas en PK - 2FN

- Ejemplo.



- ¿Está en 2FN? **¡NO!** Se ve, por DF 2 y DF 3, que hay atributos que dependen parcialmente de la PK

- Decomposición en 2FN

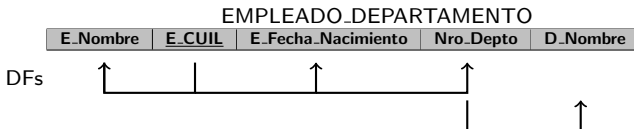


FN basadas en PK - 3FN

- **Dependencia Transitiva** Una DF $X \rightarrow Y$ en R es Transitiva, si existe un conjunto de atributos Z en R que no son ni Clave Candidata ni un subconjunto de alguna Clave de R , tal que $X \rightarrow Z$ y $Z \rightarrow Y$

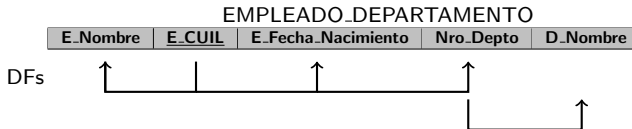
FN basadas en PK - 3FN

- **Dependencia Transitiva** Una DF $X \rightarrow Y$ en R es Transitiva, si existe un conjunto de atributos Z en R que no son ni Clave Candidata ni un subconjunto de alguna Clave de R , tal que $X \rightarrow Z$ y $Z \rightarrow Y$
- **Ejemplo.**



FN basadas en PK - 3FN

- **Dependencia Transitiva** Una DF $X \rightarrow Y$ en R es Transitiva, si existe un conjunto de atributos Z en R que no son ni Clave Candidata ni un subconjunto de alguna Clave de R , tal que $X \rightarrow Z$ y $Z \rightarrow Y$
- **Ejemplo.**

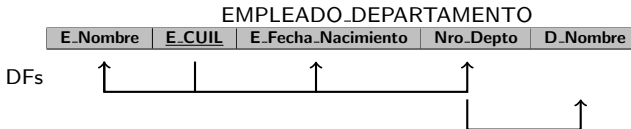


La DF $E_CUIL \rightarrow D_Nombre$ es transitiva a través de Nro_Depto ya que:

- Existe $E_CUIL \rightarrow Nro_Depto$
- Existe $Nro_Depto \rightarrow D_Nombre$
- Nro_Depto no es ni clave candidata ni parte de una clave de $EMPLEADO_DEPARTAMENTO$

FN basadas en PK - 3FN

- **Dependencia Transitiva** Una DF $X \rightarrow Y$ en R es Transitiva, si existe un conjunto de atributos Z en R que no son ni Clave Candidata ni un subconjunto de alguna Clave de R , tal que $X \rightarrow Z$ y $Z \rightarrow Y$
- **Ejemplo.**

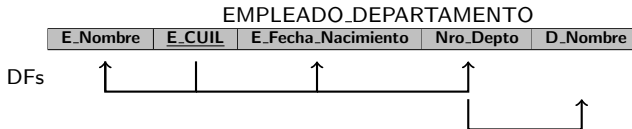


La DF $E_CUIL \rightarrow D_Nombre$ es transitiva a través de Nro_Depto ya que:

- Existe $E_CUIL \rightarrow Nro_Depto$
 - Existe $Nro_Depto \rightarrow D_Nombre$
 - Nro_Depto no es ni clave candidata ni parte de una clave de $EMPLEADO_DEPARTAMENTO$
- **3FN.** Un esquema R está en 3FN si está en 2FN y ningún atributo no primo de R depende transitivamente de la PK

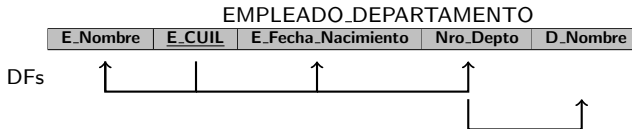
FN basadas en PK - 3FN

● Ejemplo.



FN basadas en PK - 3FN

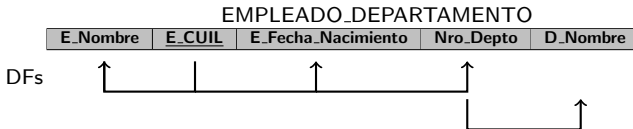
- Ejemplo.



- ¿Está en 2FN?

FN basadas en PK - 3FN

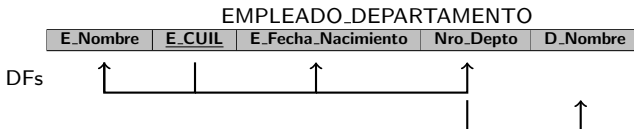
- Ejemplo.



- ¿Está en 2FN? ¡Sí! No hay dependencias parciales sobre la PK

FN basadas en PK - 3FN

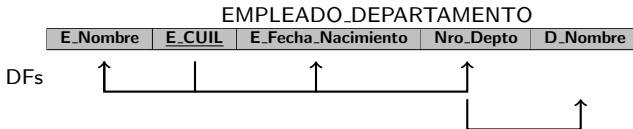
● Ejemplo.



- ¿Está en 2FN? ¡Sí! No hay dependencias parciales sobre la PK
- ¿Está en 3FN?

FN basadas en PK - 3FN

● Ejemplo.



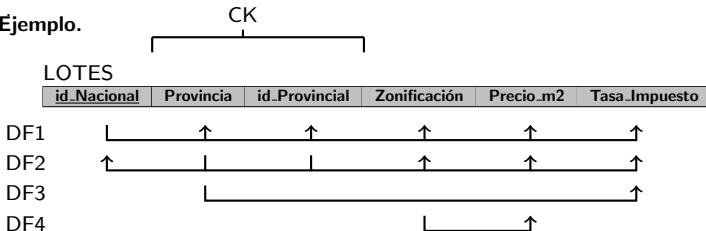
- ¿Está en 2FN? ¡Sí! No hay dependencias parciales sobre la PK
- ¿Está en 3FN? ¡NO! \exists dependencia transitiva $E_CUIL \rightarrow D_Nombre$

Definición General - 2FN

- **2FN.** Un esquema R está en 2FN si todo atributo no primo A de R no depende parcialmente (de manera funcional) de ninguna clave de R
- **2FN. Definición Alternativa.** Un esquema R está en 2FN si todo atributo no primo A de R depende completamente (de manera funcional) de todas las claves de R

Definición General - 2FN

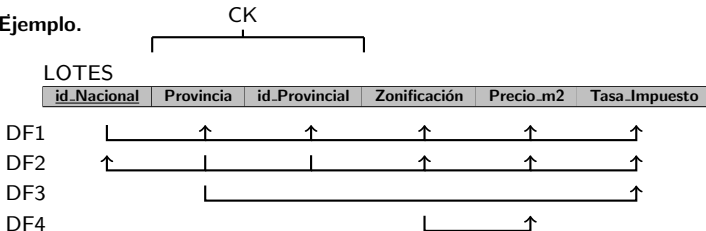
- Ejemplo.



- ¿Está en 2FN?

Definición General - 2FN

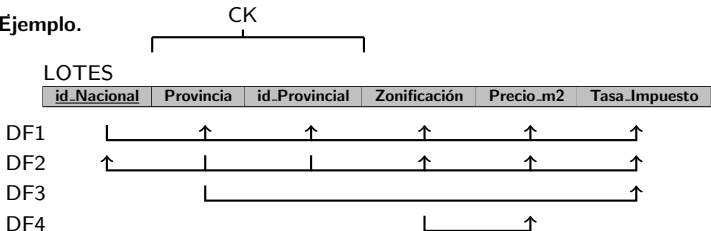
- Ejemplo.



- ¿Está en 2FN? ¡No! Tasa_Impuesto depende parcialmente de una CK (ver DF3)

Definición General - 2FN

- **Ejemplo.**



- ¿Está en 2FN? **¡No!** Tasa_Impuesto depende parcialmente de una CK (ver DF3)
- **Descomposición en 2FN.**

Definición General - 3FN

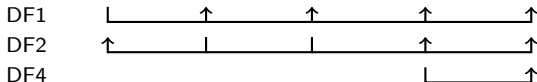
- **3FN.** Un esquema R está en 3FN si, para toda dependencia funcional *no trivial* $X \rightarrow A$ de R , se cumple alguna de las siguientes condiciones:
 - X es SK de R
 - A es atributo primo de R
- **DF trivial.** La DF $A \rightarrow B$ es trivial si B es un subconjunto de atributos de A .
Ej. $A \rightarrow A$ es una DF trivial

Definición General - 3FN

- **3FN.** Un esquema R está en 3FN si, para toda dependencia funcional *no trivial* $X \rightarrow A$ de R , se cumple alguna de las siguientes condiciones:
 - X es SK de R
 - A es atributo primo de R
- **DF trivial.** La DF $A \rightarrow B$ es trivial si B es un subconjunto de atributos de A .
Ej. $A \rightarrow A$ es una DF trivial
- **Ejemplo.**

LOTES_1

<u>id_Nacional</u>	Provincia	id_Provincial	Zonificación	Precio_m2
--------------------	-----------	---------------	--------------	-----------



LOTES_2

<u>Provincia</u>	Tasa_Impuesto
------------------	---------------



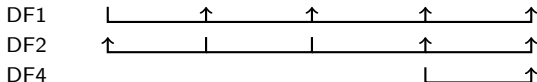
- ¿LOTES_1 está en 3FN?

Definición General - 3FN

- **3FN.** Un esquema R está en 3FN si, para toda dependencia funcional *no trivial* $X \rightarrow A$ de R , se cumple alguna de las siguientes condiciones:
 - X es SK de R
 - A es atributo primo de R
- **DF trivial.** La DF $A \rightarrow B$ es trivial si B es un subconjunto de atributos de A .
Ej. $A \rightarrow A$ es una DF trivial
- **Ejemplo.**

LOTES_1

<u>id_Nacional</u>	Provincia	id_Provincial	Zonificación	Precio_m2
--------------------	-----------	---------------	--------------	-----------



LOTES_2

<u>Provincia</u>	Tasa_Impuesto
------------------	---------------



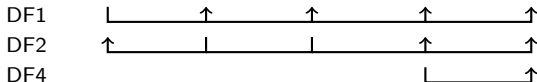
- ¿LOTES_1 está en 3FN? **¡No!** Debido a DF 3 y DF 4

Definición General - 3FN

- Ejemplo.

LOTES_1

<u>id_Nacional</u>	Provincia	id_Provincial	Zonificación	Precio_m2
--------------------	-----------	---------------	--------------	-----------

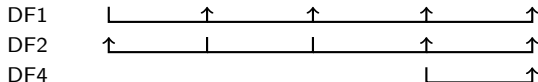


Definición General - 3FN

- Ejemplo.

LOTES_1

<u>id_Nacional</u>	Provincia	id_Provincial	Zonificación	Precio_m2
--------------------	-----------	---------------	--------------	-----------



- Descomposición en 3FN.

Definición General - BCFN (Boyce–Codd Normal Form)

● Ejemplo.

LOTES_1A

<u>id_Nacional</u>	Provincia	id_Provincial	Zonificación
--------------------	-----------	---------------	--------------

DF1 |—————↑—————↑—————↑

DF2 ↑—————|—————|—————↑

● Restricciones adicionales.

- Sólo 2 provincias: San Juan y Mendoza
- Zonificación San Juan: SJ1, SJ2, SJ3, SJ4, SJ5
- Zonificación Mendoza: MA, ME, MI, MO, MU

● Nuevo Ejemplo.

LOTES_1A

<u>id_Nacional</u>	Provincia	id_Provincial	Zonificación
--------------------	-----------	---------------	--------------

DF1 |—————↑—————↑—————↑

DF2 ↑—————|—————|—————↑

DF5 ↑—————|

● ¿Está en 3FN?

Definición General - BCFN (Boyce–Codd Normal Form)

● Ejemplo.

LOTES_1A

<u>id_Nacional</u>	Provincia	id_Provincial	Zonificación
--------------------	-----------	---------------	--------------

DF1 |—————↑—————↑—————↑

DF2 ↑—————|—————|—————↑

● Restricciones adicionales.

- Sólo 2 provincias: San Juan y Mendoza
- Zonificación San Juan: SJ1, SJ2, SJ3, SJ4, SJ5
- Zonificación Mendoza: MA, ME, MI, MO, MU

● Nuevo Ejemplo.

LOTES_1A

<u>id_Nacional</u>	Provincia	id_Provincial	Zonificación
--------------------	-----------	---------------	--------------

DF1 |—————↑—————↑—————↑

DF2 ↑—————|—————|—————↑

DF5 ↑—————|

- ¿Está en 3FN? ¡Sí! ... pero existe redundancia. Provincia se puede deducir de Zonificación

Definición General - BCFN

- Nuevo Ejemplo.

LOTES_1A

<u>id_Nacional</u>	Provincia	id_Provincial	Zonificación
--------------------	-----------	---------------	--------------



- Descomposición Boyce-Codd FN (BCFN).

Definición General - BCFN

- Nuevo Ejemplo.

LOTES_1A

<u>id_Nacional</u>	Provincia	id_Provincial	Zonificación
--------------------	-----------	---------------	--------------



- Descomposición Boyce-Codd FN (BCFN).

LOTES_1AX

<u>id_Nacional</u>	Zonificación	id_Provincial
--------------------	--------------	---------------

LOTES_1AY

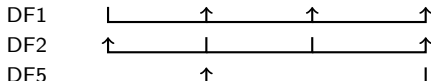
<u>Zonificación</u>	Provincia
---------------------	-----------

Definición General - BCFN

- Nuevo Ejemplo.

LOTES_1A

<u>id_Nacional</u>	Provincia	id_Provincial	Zonificación
--------------------	-----------	---------------	--------------



- Descomposición Boyce-Codd FN (BCFN).

LOTES_1AX

<u>id_Nacional</u>	Zonificación	id_Provincial
--------------------	--------------	---------------

LOTES_1AY

<u>Zonificación</u>	Provincia
---------------------	-----------

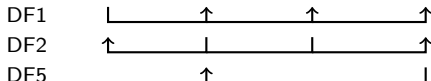
- Redundancia.** Esta representación la reduce
- Pérdida de DF.** En la descomposición se pierde DF 2 dado que sus atributos dejan de coexistir

Definición General - BCFN

- Nuevo Ejemplo.

LOTES_1A

<u>id_Nacional</u>	Provincia	id_Provincial	Zonificación
--------------------	-----------	---------------	--------------



- Descomposición Boyce-Codd FN (BCFN).

LOTES_1AX

<u>id_Nacional</u>	Zonificación	id_Provincial
--------------------	--------------	---------------

LOTES_1AY

<u>Zonificación</u>	Provincia
---------------------	-----------

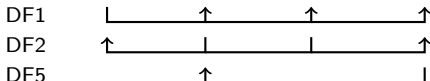
- Redundancia.** Esta representación la reduce
- Pérdida de DF.** En la descomposición se pierde DF 2 dado que sus atributos dejan de coexistir
- BCFN.** Un esquema R está en BCFN si, para toda dependencia funcional *no trivial* $X \rightarrow A$ de R , X es SK de R

Definición General - BCFN

- Nuevo Ejemplo.

LOTES_1A

<u>id_Nacional</u>	Provincia	id_Provincial	Zonificación
--------------------	-----------	---------------	--------------



- Descomposición Boyce-Codd FN (BCFN).

LOTES_1AX

<u>id_Nacional</u>	Zonificación	id_Provincial
--------------------	--------------	---------------

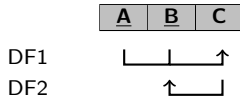
LOTES_1AY

<u>Zonificación</u>	Provincia
---------------------	-----------

- Redundancia.** Esta representación la reduce
- Pérdida de DF.** En la descomposición se pierde DF 2 dado que sus atributos dejan de coexistir
- BCFN.** Un esquema R está en BCFN si, para toda dependencia funcional *no trivial* $X \rightarrow A$ de R , X es SK de R
- BCFN vs 3FN.** BCFN es más restrictiva que 3FN ya que BCFN no permite que A sea primo

Definición General - BCFN

- **Template.** Visión esquemática de 3FN y no BCFN



Definición General - BCFN

- Ejemplo 2.

CURSADA

<u>Estudiante</u>	<u>Materia</u>	Instructor
-------------------	----------------	------------

DF1



DF2



- ¿Está en 3FN?

Definición General - BCFN

- Ejemplo 2.

CURSADA

<u>Estudiante</u>	<u>Materia</u>	Instructor
-------------------	----------------	------------

DF1



DF2



- ¿Está en 3FN? ¡Sí!

Definición General - BCFN

- Ejemplo 2.

CURSADA

<u>Estudiante</u>	<u>Materia</u>	Instructor
-------------------	----------------	------------

DF1



DF2



- ¿Está en 3FN? ¡Sí!
- ¿Está en BCFN?

Definición General - BCFN

- Ejemplo 2.

CURSADA

<u>Estudiante</u>	<u>Materia</u>	Instructor
-------------------	----------------	------------

DF1



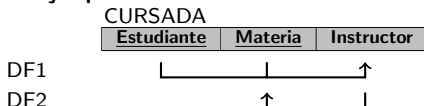
DF2



- ¿Está en 3FN? ¡Sí!
- ¿Está en BCFN? ¡No!

Definición General - BCFN

- Ejemplo 2.



- ¿Está en 3FN? ¡Sí!
- ¿Está en BCFN? ¡No!
- **Descomposición 1.** (Estudiante en ambas relaciones)

<u>Estudiante</u>	<u>Instructor</u>
-------------------	-------------------

<u>Estudiante</u>	<u>Materia</u>
-------------------	----------------

- **Descomposición 2.** (Materia en ambas relaciones)

<u>Materia</u>	<u>Instructor</u>
----------------	-------------------

<u>Materia</u>	<u>Estudiante</u>
----------------	-------------------

- **Descomposición 3.** (Instructor en ambas relaciones)

<u>Instructor</u>	<u>Materia</u>
-------------------	----------------

<u>Instructor</u>	<u>Estudiante</u>
-------------------	-------------------

Normalización - Reglas de Inferencia

- Diseñador de BD especifica DF semánticamente obvias
- Existencia de DF no especificadas que pueden ser inferidas

Normalización - Reglas de Inferencia

- Diseñador de BD especifica DF semánticamente obvias
- Existencia de DF no especificadas que pueden ser inferidas
- **Ejemplo.**
 - $R = \{E_CUIL, Nro_Depto, D_Nombre\}$
 - $F = \{E_CUIL \rightarrow Nro_Depto, Nro_Depto \rightarrow D_Nombre\}$

Normalización - Reglas de Inferencia

- Diseñador de BD especifica DF semánticamente obvias
- Existencia de DF no especificadas que pueden ser inferidas
- **Ejemplo.**
 - $R = \{E_CUIL, Nro_Depto, D_Nombre\}$
 - $F = \{E_CUIL \rightarrow Nro_Depto, Nro_Depto \rightarrow D_Nombre\}$
 - De ambas DFs se puede inferir que $E_CUIL \rightarrow D_Nombre$

Normalización - Reglas de Inferencia

- Diseñador de BD especifica DF semánticamente obvias
- Existencia de DF no especificadas que pueden ser inferidas
- **Ejemplo.**
 - $R = \{E_CUIL, Nro_Depto, D_Nombre\}$
 - $F = \{E_CUIL \rightarrow Nro_Depto, Nro_Depto \rightarrow D_Nombre\}$
 - De ambas DFs se puede inferir que $E_CUIL \rightarrow D_Nombre$
- **Inferencia.** Una DF $X \rightarrow Y$ es **inferida de** o **implicada por** un conjunto de DFs F de R si se cumple $X \rightarrow Y$ en toda instancia legal $r(R)$. Es decir, siempre que $r(R)$ satisface F , se cumple $X \rightarrow Y$

Normalización - Reglas de Inferencia

- Diseñador de BD especifica DF semánticamente obvias
- Existencia de DF no especificadas que pueden ser inferidas
- **Ejemplo.**
 - $R = \{E_CUIL, Nro_Depto, D_Nombre\}$
 - $F = \{E_CUIL \rightarrow Nro_Depto, Nro_Depto \rightarrow D_Nombre\}$
 - De ambas DFs se puede inferir que $E_CUIL \rightarrow D_Nombre$
- **Inferencia.** Una DF $X \rightarrow Y$ es **inferida de** o **implicada por** un conjunto de DFs F de R si se cumple $X \rightarrow Y$ en toda instancia legal $r(R)$. Es decir, siempre que $r(R)$ satisface F , se cumple $X \rightarrow Y$
- **Clausura.** Conjunto de todas las DFs de F más todas las DFs que puedan ser inferidas de F . Se denota como F^+
 - $R = \{E_CUIL, Nro_Depto, D_Nombre\}$
 - $F = \{E_CUIL \rightarrow Nro_Depto, Nro_Depto \rightarrow D_Nombre\}$
 - $F^+ = \{E_CUIL \rightarrow Nro_Depto, Nro_Depto \rightarrow D_Nombre, E_CUIL \rightarrow D_Nombre, \dots\}$
- **Necesidad.** Para calcular F^+ es necesario un método: Reglas de inferencia

Normalización - Reglas de Inferencia (Cont.)

- **Reglas de Inferencia.** Propuestas por Armstrong (1974) y conocidas como “**Axiomas de Armstrong**”
 - **RI1 (regla reflexiva).** Si $Y \subseteq X$, entonces $X \rightarrow Y$
 - **RI2 (regla de incremento).** $\{X \rightarrow Y\} \models XZ \rightarrow YZ$
 - **RI3 (regla transitiva).** $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$
 - **Propiedades.**
 - **Fiable (Sound).** Dado F de R , cualquier DF deducida de F utilizando RI1 a RI3, se cumple en cualquier estado $r(R)$ que satisface F
 - **Completa (Complete).** F^+ puede ser determinado a partir de F aplicando solamente RI1 a RI3

Normalización - Reglas de Inferencia (Cont.)

- **Reglas de Inferencia.** Propuestas por Armstrong (1974) y conocidas como “**Axiomas de Armstrong**”
 - **RI1 (regla reflexiva).** Si $Y \subseteq X$, entonces $X \rightarrow Y$
 - **RI2 (regla de incremento).** $\{X \rightarrow Y\} \models XZ \rightarrow YZ$
 - **RI3 (regla transitiva).** $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$
 - **Propiedades.**
 - **Fiable (Sound).** Dado F de R , cualquier DF deducida de F utilizando RI1 a RI3, se cumple en cualquier estado $r(R)$ que satisface F
 - **Completa (Complete).** F^+ puede ser determinado a partir de F aplicando solamente RI1 a RI3
- **Reglas de Inferencia Adicionales.** (corolarios de Armstrong)

Normalización - Reglas de Inferencia (Cont.)

- **Reglas de Inferencia.** Propuestas por Armstrong (1974) y conocidas como “**Axiomas de Armstrong**”
 - **RI1 (regla reflexiva).** Si $Y \subseteq X$, entonces $X \rightarrow Y$
 - **RI2 (regla de incremento).** $\{X \rightarrow Y\} \models XZ \rightarrow YZ$
 - **RI3 (regla transitiva).** $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$
 - **Propiedades.**
 - **Fiable (Sound).** Dado F de R , cualquier DF deducida de F utilizando RI1 a RI3, se cumple en cualquier estado $r(R)$ que satisface F
 - **Completa (Complete).** F^+ puede ser determinado a partir de F aplicando solamente RI1 a RI3
- **Reglas de Inferencia Adicionales.** (corolarios de Armstrong)
 - **RI4 (regla de descomposición o proyección).**
 $\{X \rightarrow YZ\} \models X \rightarrow Y$
 - **RI5 (regla de unión o aditiva).** $\{X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$
 - **RI6 (regla pseudotransitiva).** $\{X \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z$

Normalización - Clausura

- **Diseño.** Típicamente

- 1 Diseñador especifica conjunto de DFs F determinadas por semántica de atributos de R
- 2 Se utilizan RI1 a RI3 para inferir DFs adicionales

Normalización - Clausura

- **Diseño.** Típicamente
 - 1 Diseñador especifica conjunto de DFs F determinadas por semántica de atributos de R
 - 2 Se utilizan RI1 a RI3 para inferir DFs adicionales
- ¿Cómo realizar (2) de manera sistemática?

Normalización - Clausura

- **Diseño.** Típicamente
 - 1 Diseñador especifica conjunto de DFs F determinadas por semántica de atributos de R
 - 2 Se utilizan RI1 a RI3 para inferir DFs adicionales
- ¿Cómo realizar (2) de manera sistemática?
 - determinar conjunto de atributos X que aparecen del lado izq. de DFs de F
 - determinar conjunto Y de todos los atributos que dependen de X

Normalización - Clausura

- **Diseño.** Típicamente
 - 1 Diseñador especifica conjunto de DFs F determinadas por semántica de atributos de R
 - 2 Se utilizan RI1 a RI3 para inferir DFs adicionales
- ¿Cómo realizar (2) de manera sistemática?
 - determinar conjunto de atributos X que aparecen del lado izq. de DFs de F
 - determinar conjunto Y de todos los atributos que dependen de X
- **Clausura de X .** Conjunto de atributos que son determinados por X basados en F . Se nota X^+

Normalización - Clausura

- **Diseño.** Típicamente
 - 1 Diseñador especifica conjunto de DFs F determinadas por semántica de atributos de R
 - 2 Se utilizan RI1 a RI3 para inferir DFs adicionales
- ¿Cómo realizar (2) de manera sistemática?
 - determinar conjunto de atributos X que aparecen del lado izq. de DFs de F
 - determinar conjunto Y de todos los atributos que dependen de X
- **Clausura de X .** Conjunto de atributos que son determinados por X basados en F . Se nota X^+
- **Algoritmo Nro. 1** para determinar X^+

Entrada: DFs F de R ; subconjunto de atributos X de R

1. $X^+ := X$
2. **repetir**
3. $\text{viejo}X^+ := X^+$
4. **Para cada DF** $Y \rightarrow Z$ **en** F **hacer**
5. **Si** $Y \subseteq \text{viejo}X^+$ **entonces** $X^+ = X^+ \cup Z$
6. **hasta** ($X^+ = \text{viejo}X^+$)

Normalización - Clausura (Cont.)

● Ejemplo.

- $R = (idClase, CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap)$
- $F = \{$
 - $DF1: idClase \rightarrow \{ CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \},$
 - $DF2: CodCurso \rightarrow Puntos,$
 - $DF3: \{ CodCurso, Instr \} \rightarrow \{ Libro, Aula \},$
 - $DF4: Libro \rightarrow Ed,$
 - $DF5: Aula \rightarrow Cap$ $\}$

Normalización - Clausura (Cont.)

- **Ejemplo.**

- $R = (idClase, CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap)$
- $F = \{$
 - $DF1: idClase \rightarrow \{ CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \},$
 - $DF2: CodCurso \rightarrow Puntos,$
 - $DF3: \{ CodCurso, Instr \} \rightarrow \{ Libro, Aula \},$
 - $DF4: Libro \rightarrow Ed,$
 - $DF5: Aula \rightarrow Cap$ $\}$

- Aplicando el algoritmo para obtener x^+

- $\{ idClase \}^+ =$

Normalización - Clausura (Cont.)

- **Ejemplo.**

- $R = (idClase, CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap)$
- $F = \{$
 - $DF1: idClase \rightarrow \{ CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \},$
 - $DF2: CodCurso \rightarrow Puntos,$
 - $DF3: \{ CodCurso, Instr \} \rightarrow \{ Libro, Aula \},$
 - $DF4: Libro \rightarrow Ed,$
 - $DF5: Aula \rightarrow Cap$ $\}$

- Aplicando el algoritmo para obtener X^+

- $\{ idClase \}^+ = \{ idClase, CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \} = R$

Normalización - Clausura (Cont.)

- **Ejemplo.**

- $R = (idClase, CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap)$
- $F = \{$
 - $DF1: idClase \rightarrow \{ CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \},$
 - $DF2: CodCurso \rightarrow Puntos,$
 - $DF3: \{ CodCurso, Instr \} \rightarrow \{ Libro, Aula \},$
 - $DF4: Libro \rightarrow Ed,$
 - $DF5: Aula \rightarrow Cap$ $\}$

- **Aplicando el algoritmo para obtener x^+**

- $\{ idClase \}^+ = \{ idClase, CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \} = R$
- $\{ CodCurso \}^+ =$

Normalización - Clausura (Cont.)

- **Ejemplo.**

- $R = (idClase, CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap)$
- $F = \{$
 - $DF1: idClase \rightarrow \{ CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \},$
 - $DF2: CodCurso \rightarrow Puntos,$
 - $DF3: \{ CodCurso, Instr \} \rightarrow \{ Libro, Aula \},$
 - $DF4: Libro \rightarrow Ed,$
 - $DF5: Aula \rightarrow Cap$ $\}$

- **Aplicando el algoritmo para obtener x^+**

- $\{ idClase \}^+ = \{ idClase, CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \} = R$
- $\{ CodCurso \}^+ = \{ CodCurso, Puntos \}$

Normalización - Clausura (Cont.)

- **Ejemplo.**

- $R = (idClase, CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap)$
- $F = \{$
 - $DF1: idClase \rightarrow \{ CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \},$
 - $DF2: CodCurso \rightarrow Puntos,$
 - $DF3: \{ CodCurso, Instr \} \rightarrow \{ Libro, Aula \},$
 - $DF4: Libro \rightarrow Ed,$
 - $DF5: Aula \rightarrow Cap$ $\}$

- **Aplicando el algoritmo para obtener x^+**

- $\{ idClase \}^+ = \{ idClase, CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \} = R$
- $\{ CodCurso \}^+ = \{ CodCurso, Puntos \}$
- $\{ CodCurso, Instr \}^+ =$

Normalización - Clausura (Cont.)

- **Ejemplo.**

- $R = (idClase, CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap)$
- $F = \{$
 - $DF1: idClase \rightarrow \{ CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \},$
 - $DF2: CodCurso \rightarrow Puntos,$
 - $DF3: \{ CodCurso, Instr \} \rightarrow \{ Libro, Aula \},$
 - $DF4: Libro \rightarrow Ed,$
 - $DF5: Aula \rightarrow Cap$ $\}$

- **Aplicando el algoritmo para obtener x^+**

- $\{ idClase \}^+ = \{ idClase, CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \} = R$
- $\{ CodCurso \}^+ = \{ CodCurso, Puntos \}$
- $\{ CodCurso, Instr \}^+ = \{ CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \}$

Normalización - Clausura (Cont.)

- **Ejemplo.**

- $R = (idClase, CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap)$
- $F = \{$
 - $DF1: idClase \rightarrow \{ CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \},$
 - $DF2: CodCurso \rightarrow Puntos,$
 - $DF3: \{ CodCurso, Instr \} \rightarrow \{ Libro, Aula \},$
 - $DF4: Libro \rightarrow Ed,$
 - $DF5: Aula \rightarrow Cap$ $\}$

- Aplicando el algoritmo para obtener x^+

- $\{ idClase \}^+ = \{ idClase, CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \} = R$
- $\{ CodCurso \}^+ = \{ CodCurso, Puntos \}$
- $\{ CodCurso, Instr \}^+ = \{ CodCurso, Instr, Puntos, Libro, Ed, Aula, Cap \}$

- **Observación.** Clausura $idClase \not\subseteq \{ CodCurso, Instr \}^+$ por lo tanto NO es CK

Normalización - Clave de una Relación

Algoritmo Búsqueda de una clave K de R a partir de un conjunto de DFs

Normalización - Clave de una Relación

Algoritmo Búsqueda de una clave K de R a partir de un conjunto de DFs

Entrada: Relación R y un Conjunto de DFs F de R

1. $K := R$
2. Para cada atributo $A \in K$
 Computar $(K - A)^+$ con respecto a F
 Si $(K - A)^+$ contiene todos los atributos de R entonces $K := K - \{A\}$

Normalización - Clave de una Relación

Algoritmo Búsqueda de una clave K de R a partir de un conjunto de DFs

Entrada: Relación R y un Conjunto de DFs F de R

1. $K := R$
2. Para cada atributo $A \in K$
 Computar $(K - A)^+$ con respecto a F
 Si $(K - A)^+$ contiene todos los atributos de R entonces $K := K - \{A\}$

- Algoritmo determina una sola de las CK. Depende fuertemente de la manera en que son removidos los atributos

Normalización - Bibliografía

- Capítulo 15 (hasta 15.5 inclusive) Elmasri/Navathe - Fundamentals of Database Systems, 6th Ed., Pearson, 2011.

