

TUIA - BASES DE DATOS I

NORMALIZACIÓN – 1FN - REPASO

Definición:

Una relación está en la primera forma Normal (1FN) si y sólo si cada atributo tiene un valor atómico para cada tupla.

TUIA - BASES DE DATOS I

NORMALIZACIÓN – 1FN - EJEMPLO - REPASO

PAC_CONS	<u>NroHist</u>	NomPac	CodOS	Descrip	<u>CodMed</u>	NomMed	<u>Fecha</u>	Diagnostico
	1912	Gomez, A.	2	Osecac	101	Perez, J.	23/01/2001	Angina
	1912	Gomez, A.	2	Osecac	103	Sosa, T.	24/02/2001	Otitis
	1912	Gomez, A.	2	Osecac	104	Juárez, I.	01/04/2002	Hepatitis
	8123	Estévez, H.	3	Osde	103	Sosa, T.	30/08/2001	Angina
	5407	Alvarez, J.	1	OsUNR	103	Sosa, T.	04/07/2002	Gripe
	5407	Alvarez, J.	1	OsUNR	103	Sosa, T.	09/09/2002	Bronquitis
	7191	Benitez, A.	2	Osecac	102	Tapia, D.	12/06/2001	Gripe
	7191	Benitez, A.	2	Osecac	104	Juárez, I.	31/01/2002	Angina

Estrictamente está en 1FN:

- Pero posee redundancia de datos/posibles anomalías de actualización.

TUIA - BASES DE DATOS I

NORMALIZACIÓN – 2FN - EJEMPLO - REPASO

- Una DF es una **dependencia funcional total** si la eliminación de cualquier atributo A de X hace que la dependencia deje de ser válida.
- Una DF es una **dependencia funcional parcial** si es posible eliminar un atributo A de X y la dependencia sigue siendo válida.

TUIA - BASES DE DATOS I

NORMALIZACIÓN – 2FN - REPASO

Definición:

Una relación está en segunda forma Normal (2FN) si y sólo si está en 1FN y todos los atributos no clave son completamente dependientes de la clave.

TUIA - BASES DE DATOS I

NORMALIZACIÓN – 2FN - REPASO

MEDICOS	<u>CodMed</u>	NomMed
	101	Perez, J.
	103	Sosa, T.
	104	Juárez, I.
	102	Tapia, D.

PAC_OBRASOC	<u>NroHist</u>	NomPac	CodOS	Descrip
	1912	Gomez, A.	2	Osecac
	8123	Estévez, H.	3	Osde
	5407	Alvarez, J.	1	OsUNR
	7191	Benitez, A.	2	Osecac

CONSULTAS	<u>NroHist</u>	<u>CodMed</u>	<u>Fecha</u>	Diagnostico
	1912	101	23/01/2001	Angina
	1912	103	24/02/2001	Otitis
	1912	104	01/04/2002	Hepatitis
	8123	103	30/08/2001	Angina
	5407	103	04/07/2002	Gripe
	5407	103	09/09/2002	Bronquitis
	7191	102	12/06/2001	Gripe
	7191	104	31/01/2002	Angina

TUIA - BASES DE DATOS I

NORMALIZACIÓN – 3FN - REPASO

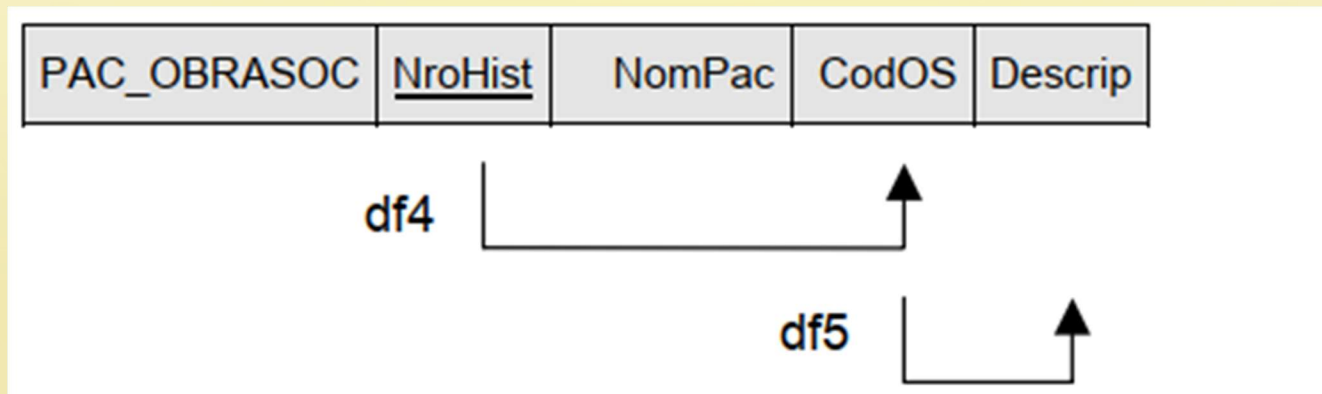
Dependencia transitiva:

Una dependencia $X \rightarrow Y$ es una dependencia funcional transitiva si existe un conjunto de atributos Z que no sea parte de cualquier clave y se cumplen tanto $X \rightarrow Z$ como $Z \rightarrow Y$

TUIA - BASES DE DATOS I

NORMALIZACIÓN – 3FN - REPASO

Dependencia transitiva:



TUIA - BASES DE DATOS I

NORMALIZACIÓN – 3FN - REPASO

Definición:

Una relación está en tercera forma Normal (3FN) si y sólo si está en 2FN y todos los atributos no clave dependen de manera no transitiva de la clave primaria.

TUIA - BASES DE DATOS I

NORMALIZACIÓN – 3FN - EJEMPLO - REPASO

A lo que llegamos:

PACIENTES	<u>NroHist</u>	NomPac	CodOS
	1912	Gomez, A.	2
	8123	Estévez, H.	3
	5407	Alvarez, J.	1
	7191	Benitez, A.	2

OBRASOC	<u>CodOS</u>	Descrip
	2	Osecac
	3	Osde
	1	OsUNR

CONSULTAS	<u>NroHist</u>	<u>CodMed</u>	<u>Fecha</u>	Diagnostico
	1912	101	23/01/2001	Angina
	1912	103	24/02/2001	Otitis
	1912	104	01/04/2002	Hepatitis
	8123	103	30/08/2001	Angina
	5407	103	04/07/2002	Gripe
	5407	103	09/09/2002	Bronquitis
	7191	102	12/06/2001	Gripe
	7191	104	31/01/2002	Angina

MEDICOS	<u>CodMed</u>	NomMed
	101	Perez, J.
	103	Sosa, T.
	104	Juárez, I.
	102	Tapia, D.

TUIA - BASES DE DATOS I

JOIN

Permite combinar campos de 2 tablas. Ejemplo de sintaxis “tradicional” (antes que exista el estándar ANSI) usando la Base de Datos Northwind:

```
SELECT *  
FROM Customers C, Orders O  
WHERE C.CustomerID = O.CustomerID
```

TUIA - BASES DE DATOS I

JOIN

El ejemplo anterior es una forma de escribir una consulta de combinación interna (INNER JOIN) en la sintaxis de **SQL SERVER**.

Hay otra forma de escribirla que corresponde al estándar **ANSI** y es más habitual...

TUIA - BASES DE DATOS I

JOIN

Clausula **JOIN**

La cláusula **JOIN** nos permite combinar 2 o más tablas en base a una condición

TUIA - BASES DE DATOS I

JOIN

- Especificar las condiciones de la combinación en la cláusula **FROM** ayuda a separarlas de cualquier otra condición de búsqueda que se pueda especificar en una cláusula **WHERE**.
- Es el método *recomendado* para especificar combinaciones. La sintaxis simplificada de combinación de la cláusula **FROM** de ANSI es:

TUIA - BASES DE DATOS I

JOIN – SINTAXIS ANSI

```
FROM first_table  
    <join_type> second_table  
        [ON (join_condition)]
```

TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

INNER JOIN (combinación interna)

Realiza el producto cartesiano y luego filtra según la cláusula ON

TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

INNER JOIN (combinación interna)

La palabra clave INNER es opcional.

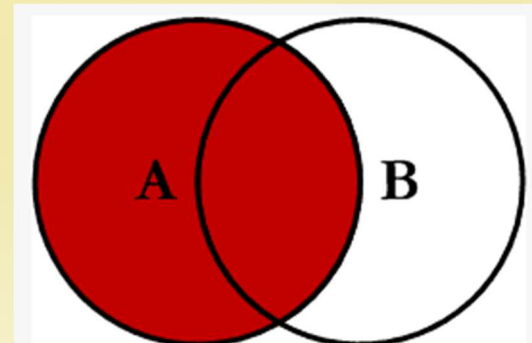
```
SELECT *  
FROM Customers C  
    INNER JOIN Orders O  
    ON C.CustomerID = O.CustomerID
```

TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

LEFT JOIN (combinación externa izquierda)

- Combina todos los registros de la **primer** tabla con los de la **segunda** que cumplan la condición.
- Cuando no haya coincidencia en la segunda tabla se completa con NULOS.



TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

LEFT JOIN (combinación externa izquierda)

```
SELECT *  
FROM Customers C  
    LEFT JOIN Orders O  
    ON C.CustomerID = O.CustomerID
```

TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

LEFT JOIN (combinación externa izquierda)

La palabra clave **OUTER** es opcional

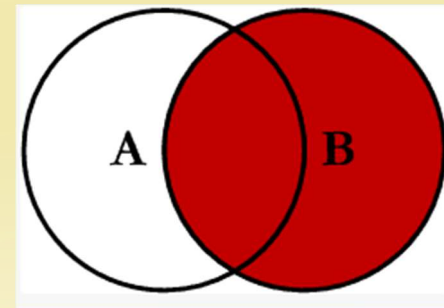
```
SELECT *  
FROM Customers C  
    LEFT OUTER JOIN Orders O  
    ON C.CustomerID = O.CustomerID
```

TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

RIGHT JOIN (combinación externa derecha)

- Combina todos los registros de la **segunda** tabla con los de la **primera** que cumplan la condición.
- Cuando no haya coincidencia en la primera tabla se completa con NULOS.



TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

RIGHT JOIN (combinación externa derecha)

La palabra clave *OUTER* es opcional

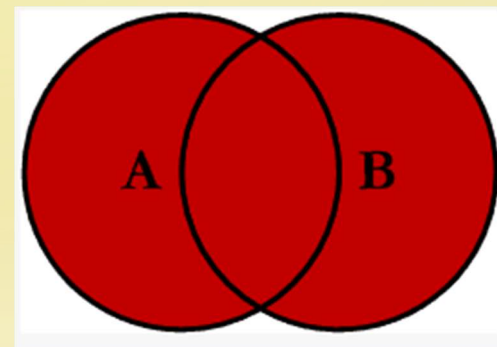
```
SELECT *  
FROM Orders O  
      RIGHT OUTER JOIN Customers C  
      ON C.CustomerID = O.CustomerID
```

TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

FULL JOIN (combinación externa completa)

- Devuelve un conjunto de resultados que incluye filas de las tablas izquierda y derecha.



TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

FULL JOIN (combinación externa completa)

- Cuando no existen filas coincidentes para la fila de la tabla de la izquierda, las columnas de la tabla de la derecha tendrán nulos.
- Cuando no existen filas coincidentes para la fila de la tabla de la derecha, las columnas de la tabla de la izquierda tendrán NULL

TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

FULL JOIN (combinación externa completa)

La palabra clave OUTER es opcional

```
SELECT *  
FROM Customers C  
      FULL OUTER JOIN Orders O  
      ON C.CustomerID = O.CustomerID
```

TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

JOIN con USING (no disponible en SQL Server)

Permite especificar una única vez la o las columnas que deben igualarse y devuelve la(s) mismas una única vez.

```
SELECT * FROM Customers C  
    [INNER|LEFT|RIGHT|FULL] JOIN Orders O  
    USING (CustomerID)
```

Nota: La sintaxis de USING se presenta sólo a título informativo y no será evaluado

TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

JOIN con NATURAL (no disponible en SQL Server)

No requiere especificar las columnas a igualarse, las detecta en base a aquellas con nombres iguales en ambas tablas.

```
SELECT *  
FROM Customers C  
    NATURAL [INNER|LEFT|RIGHT|FULL] JOIN Orders O;
```

Nota: La sintaxis de NATURAL JOIN se presenta sólo a título informativo y no será evaluado

TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

CROSS JOIN (Producto Cartesiano)

No tiene una condición ya que deseamos combinar todas las filas de la primer tabla con todas las de la segunda.

```
SELECT *  
FROM Customers C  
      CROSS JOIN Orders O
```

TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

CROSS JOIN (Producto Cartesiano)

Se puede obtener los mismos resultados usando la sintaxis tradicional (no ANSI) sin una condición de WHERE:

```
SELECT *  
FROM Customers C, Orders O;
```

Pero no es tan claro como el CROSS JOIN que se desea hacer un producto cartesiano

TUIA - BASES DE DATOS I

JOIN – TIPOS DE JOIN

CROSS JOIN (Producto Cartesiano)

Se puede obtener los mismos resultados usando la sintaxis ANSI de JOIN con un ON que devuelva siempre verdadero:

```
SELECT *  
FROM Customers C  
      JOIN Orders O ON 1=1;
```