

Bases de Datos 1

Alejandra Lliteras

alejandra.lliteras@lifa.info.unlp.edu.ar



En la clase anterior...

**Ejercicio para entregar el
miércoles 13/09 como parte del
régimen de promoción.**

**El parcialito de promoción, es el
miércoles 20/09.**

Ejercicio para entregar

Lugar_trabajo (empleado, departamento)

Curso_departamento (departamento, curso)

Curso_realizado (empleado, curso)

b) ¿Qué empleados hicieron todos los cursos requeridos por su departamento?

En esta clase...

Ejercicio

Lugar_trabajo (empleado, departamento)

Curso_departamento (departamento, curso)


Curso_realizado (empleado, curso)

b) ¿Qué empleados hicieron todos los cursos requeridos por su departamento?

Los cursos que tienen que hacer los empleados:

Los empleados y los cursos que no hicieron:

Los empleados que hicieron todos los cursos exigidos por su departamento:



Ejercicio

Lugar_trabajo (empleado, departamento)

Curso_departamento (departamento, curso)

Curso_realizado (empleado, curso)

b) ¿Qué empleados hicieron todos los cursos requeridos por su departamento?

Los cursos que tienen que hacer los empleados:

$$A = \Pi_{\text{empleado, curso}} (\text{Lugar_trabajo} \mid X \mid \text{Curso_departamento})$$

Los empleados y los cursos que no hicieron:

$$B = (A) - (\text{Curso_realizado})$$

Los empleados que hicieron todos los cursos exigidos por su departamento:

$$\Pi_{\text{empleado}} (\text{Lugar_trabajo}) - \Pi_{\text{empleado}} (B)$$

Ejercicio

Empleado(dni, nombre, domicilio, salario)

¿Qué empleados cobran el máximo valor de salario?



Ejercicio

Empleado(dni, nombre, domicilio, salario)

¿Qué empleados cobran el máximo valor de salario?

- **Buscar el valor del salario máximo**
 - **Busco los salarios que están por debajo del resto**
 - **A todos los salarios le resto los mínimos**
- **Relaciono el máximo salario con las personas que lo perciben**

Empleado(dni, nombre, domicilio, salario)

¿Qué empleados cobran el máximo valor de salario?

- **Buscar el valor del salario máximo**
 - **Busco los salarios que están por debajo del resto**

$\rho_{EMP(dni,nom,sal)}$ (Empleado)

$EMPLE \leftarrow \Pi_{dni, nombre, salario} (Empleado)$

$EmplBajoSalario \leftarrow \sigma_{salario < sal} (EMPLE \times EMP)$

$SalarioDebajoMaximo \leftarrow \Pi_{salario} (EmplBajoSalario)$

EMPLE

dni	nombre	salario
123456	Juan	10200
456789	Pedro	15000
567890	María	22000
234567	Joaquín	28000
345678	Martina	25000
678901	Mario	28000

EMP

dni	nom	sal
123456	Juan	10200
456789	Pedro	15000
567890	María	22000
234567	Joaquín	28000
345678	Martina	25000
678901	Mario	28000

EmplBajoSalario ←

$\sigma_{emple.salario < sal(EMPLEX EMP)}$

EMPLE

dni	nombre	salario
123456	Juan	10200
456789	Pedro	15000
567890	María	22000
234567	Joaquín	28000
345678	Martina	25000
678901	Mario	28000

EMP

dni	nom	sal
123456	Juan	10200
456789	Pedro	15000
567890	María	22000
234567	Joaquín	28000
345678	Martina	25000
678901	Mario	28000

EmplBajoSalario

EMPLE.dni	nombre	salario	EMP.dni	nom	sal
123456	Juan	10200	456789	Pedro	15000
123456	Juan	10200	567890	María	22000
123456	Juan	10200	234567	Joaquín	28000
123456	Juan	10200	345678	Martina	25000
123456	Juan	10200	678901	Mario	28000
456789	Pedro	15000	567890	María	22000
456789	Pedro	15000	234567	Joaquín	28000
456789	Pedro	15000	345678	Martina	25000
456789	Pedro	15000	678901	Mario	28000
567890	María	22000	234567	Joaquín	28000
567890	María	22000	345678	Martina	25000
567890	María	22000	678901	Mario	28000
345678	Martina	25000	234567	Joaquín	28000
345678	Martina	25000	678901	Mario	28000

SalarioDebajoMaximo $\leftarrow \Pi_{\text{salario}} (\text{EmplBajoSalario})$

EmplBajoSalario

EMPLE.dni	nombre	salario	EMP.dni	nom	sal
123456	Juan	10200	456789	Pedro	15000
123456	Juan	10200	567890	María	22000
123456	Juan	10200	234567	Joaquín	28000
123456	Juan	10200	345678	Martina	25000
123456	Juan	10200	678901	Mario	28000
456789	Pedro	15000	567890	María	22000
456789	Pedro	15000	234567	Joaquín	28000
456789	Pedro	15000	345678	Martina	25000
456789	Pedro	15000	678901	Mario	28000
567890	María	22000	234567	Joaquín	28000
567890	María	22000	345678	Martina	25000
567890	María	22000	678901	Mario	28000
345678	Martina	25000	234567	Joaquín	28000
345678	Martina	25000	678901	Mario	28000

SalarioDebajoMaximo

salario
10200
15000
22000
25000

Empleado(dni, nombre, domicilio, salario)

¿Qué empleados cobran el máximo valor de salario?

- **Buscar el valor del salario máximo**
 - Busco los salarios que están por debajo del resto
 - **A todos los salarios le resto los mínimos**

MaximoSalario $\leftarrow \Pi_{\text{salario}}(\text{Empleado}) - \text{SalarioDebajoMaximo}$

$\text{maximoSalario} \leftarrow \Pi_{\text{salario}}(\text{Empleado}) - \text{salarioDebajoMaximo}$

$\Pi_{\text{salario}}(\text{Empleado})$

salario
10200
15000
22000
28000
25000

SalarioDebajoMaximo

salario
10200
15000
22000
25000

MaximoSalario

salario
28000

Empleado(dni, nombre, domicilio, salario)

¿Qué empleados cobran el máximo valor de salario?

- **Buscar el valor del salario máximo**
 - Busco los salarios que están por debajo del resto
 - A todos los salarios le resto los mínimos
- **Relaciono el máximo salario con las personas que lo perciben**

Resultado ← Empleado |X| MaximoSalario

Empleado(dni, nombre, domicilio, salario)

¿Qué empleados cobran el máximo valor de salario?

- Buscar el valor del salario máximo
 - Busco los salarios que están por debajo del resto
 - A todos los salarios le resto los mínimos
- Relaciono el máximo salario con las personas que lo perciben

Resultado ← **Empleado** |X| **MaximoSalario**

Empleado

dni	nombre	domicilio	salario
123456	Juan	1 y 36	10200
456789	Pedro	154 y 78	15000
567890	María	22 y 61	22000
234567	Joaquín	12 y 62	28000
345678	Martina	58 y 7	25000
678901	Mario	120 y 50	28000

MaximoSalario

salario
28000

Resultado

dni	nombre	domicilio	salario
234567	Joaquín	12 y 62	28000
678901	Mario	120 y 50	28000

Teoría de diseño de BBDD relacionales



Teoría de diseño de BBDD relacionales

- Relación
- Anomalía
- Dependencia Funcional
- Dependencia Funcional Trivial
- Clave
 - Clave Candidata
 - Super Clave
- Axiomas de Armstrong
- Clausura de un conjunto de atributos

Teoría de diseño de BBDD relacionales

► Modelo Relacional

◦ Relación

- ¿Qué criterio uso para crear una relación?
- ¿Cuándo una relación esta bien diseñada?
- ¿Qué propiedades debería cumplir?

Teoría de diseño de BBDD relacionales

► Anomalía

- Problema que surge a raíz del diseño de una relación

Teoría de diseño de BBDD relacionales

► Anomalías

- **Redundancia:** Información que se repite innecesariamente en diferentes tuplas
- **Anomalías de actualización:** Se puede actualizar el valor en una tupla, sin actualizar los de otras tuplas
- **Anomalías de inserción:** insertar valores en ciertos atributos de una relación y no en otros me produce valores nulos
- **Anomalías de borrado:** borrar ciertos valores de una tupla, puede llevarme a perder la información de la tupla completa

Teoría de diseño de BBDD relacionales

PERSONAEMPLEADA(dni, nombre, domicilio, depto, flngDepto, codEmpDepto, jefe)

dni	nombre	domicilio	depto	flngDepto	codEmpDepto	jefe
dni1	Juan	12 Nro 222	Compras	2016-01-11	E1	J1
dni1	Juan	12 Nro 222	Liq. Sueldos	2015-02-01	E2	J2
dni2	Maria	3 Nro. 214	Compras	2014-05-01	E3	J1
dni3	José	3 Nro. 214	Compras	2015-01-08	E4	J1

El código de jefe es único en el sistema. Una persona puede trabajar en mas de un departamento y en cada uno de ellos posee un código de empleado. El código de empleado no se repite ni por departamento, ni entre departamentos.

Teoría de diseño de BBDD relacionales

► Anomalías

- **Redundancia:** Información que se repite innecesariamente en diferentes tuplas

dni	nombre	domicilio	depto	fIngDepto	codEmpDepto	jefe
dni 1	Juan	12 Nro 222	Compras	2016-01-11	E1	J1
dni 1	Juan	12 Nro 222	Liq. Sueldos	2015-02-01	E2	J2
dni 2	Maria	3 Nro. 214	Compras	2014-05-01	E3	J1
dni 3	José	3 Nro. 214	Compras	2015-01-08	E4	J1

Teoría de diseño de BBDD relacionales

► Anomalías

- **Anomalías de actualización:** Se puede actualizar el valor en una tupla, sin actualizar los de otras tuplas

dni	nombre	domicilio	depto	fIngDepto	codEmpDepto	jefe
dni 1	Juan	10 Nro 222	Compras	2016-01-11	E1	J1
dni 1	Juan	12 Nro 222	Liq. Sueldos	2015-02-01	E2	J2
dni 2	Maria	3 Nro. 214	Compras	2014-05-01	E3	J1
dni 3	José	3 Nro. 214	Compras	2015-01-08	E4	J1

Teoría de diseño de BBDD relacionales

► Anomalías

- **Anomalías de inserción:** insertar valores en ciertos atributos de una relación y no en otros me produce valores nulos

dni	nombre	domicilio	depto	fIngDepto	codEmpDepto	jefe
dni 1	Juan	12 Nro 222	Compras	2016-01-11	E1	J1
dni 1	Juan	12 Nro 222	Liq. Sueldos	2015-02-01	E2	J2
dni 2	Maria	3 Nro. 214	Compras	2014-05-01	E3	J1
dni 3	José	3 Nro. 214	Compras	2015-01-08	E4	J1

¿Qué sucede si quiero insertar solamente datos de la persona y aun no lo tengo asignado a un departamento?

Teoría de diseño de BBDD relacionales

► Anomalías

- **Anomalías de borrado:** borrar ciertos valores de una tupla, puede llevarme a perder la información de la tupla completa

dni	nombre	domicilio	depto	fIngDepto	codEmpDepto	jefe
dni 1	Juan	12 Nro 222	Compras	2016-01-11	E1	J1
dni 1	Juan	12 Nro 222	Liq. Sueldos	2015-02-01	E2	J2
dni 2	Maria	3 Nro. 214	Compras	2014-05-01	E3	J1
dni 3	José	3 Nro. 214	Compras	2015-01-08	E4	J1

¿Qué sucede si quiero borrar solamente el dato del jefe?

Teoría de diseño de BBDD relacionales

PERSONAEMPLEADA(dni, nombre, domicilio, depto,
flngDepto, codEmpDepto, jefe)

¿Es un buen diseño?



Teoría de diseño para bases de datos relaciones

DEPENDENCIA FUNCIONAL



Teoría de diseño de BBDD relacionales

► Dependencia funcional:

- Captura propiedades del mundo real
- Es una **restricción** de una relación en una base de datos
- Generaliza la idea de clave de una relación

Teoría de diseño de BBDD relacionales

► Dependencia Funcional (df)

- Si dos tuplas (t_1 y t_2) de una relación R , coinciden en todos los atributos A_1, A_2, \dots, A_n , entonces DEBEN también coincidir en los atributos B_1, B_2, \dots, B_m . Para toda tupla de R .

- Esto se escribe

$$A_1, A_2, \dots, A_n \rightarrow B_1, B_2, \dots, B_m$$

- Y se lee

A_1, A_2, \dots, A_n “determina funcionalmente a” B_1, B_2, \dots, B_m

Cuando R cumple una df, estamos indicando una restricción sobre toda la relación R y no sobre algunas tuplas de R .

Teoría de diseño de BDD relacionales

► Dependencia Funcional (df)

- Dicho de otra manera:
 - Una dependencia funcional de la forma $X \rightarrow Y$ se cumple en R si:
 - Para todos los pares de tuplas $t1$ y $t2$ de la relación, cuando se cumple que $t1[x]=t2[x]$,
 - entonces se cumple $t1[y]=t2[y]$.

Teoría de diseño de BBDD relacionales

► Dependencia Funcional (df)

- Ejemplos:

- Dada la relación: PERSONA(dni, nombre, edad, fechaNacimiento)
- Y valga en PERSONA la df: $\text{dni} \rightarrow \text{nombre, edad, fechaNac}$
 - La df enunciada, indica que si dos tuplas t1 y t2 de la relación PERSONA tienen el mismo valor en el atributo dni, deben necesariamente tener los mismos valores en los atributos nombre, edad y fechaNac

Teoría de diseño de BBDD relacionales

► Dependencia Funcional (df)

- Ejemplos:

- Dada la relación: PERSONA(dni, nombre, edad, fechaNacimiento, nroLegajo)
- *Donde*
 - *Una persona posee un único número de legajo asignado*
 - *Un número de legajo pertenece a una sola persona*

- Se pueden enunciar las siguientes dfs

df1) dni \rightarrow nombre, edad, fechaNac, nroLegajo

df2) nroLegajo \rightarrow nombre, edad, fechaNac, dni

Teoría de diseño de BBDD relacionales

► Dependencia Funcional (df)

◦ Ejemplos:

- Dada la relación: PERSONA(dni, nombre, edad, fechaNacimiento, nroLegajo, carrera)
 - *Donde*
 - *Una persona puede cursar diversas carreras*
 - *Nombre indica como se llama la persona*
 - *Una persona posee un único número de legajo asignado para cada carrera que cursa*
 - *Un número de legajo pertenece a una sola persona de una carrera*

◦ Se pueden enunciar las siguientes dfs

df1) dni \rightarrow nombre, edad, fechaNac

df2) nroLegajo, carrera \rightarrow dni

df3) dni, carrera \rightarrow nroLegajo

Teoría de diseño para bases de datos relaciones

DEPENDENCIA FUNCIONAL TRIVIAL



Teoría de diseño de BBDD relacionales

► Dependencia Funcional trivial

Es una df de la forma:

$$A_1, A_2, \dots, A_n \rightarrow B_1, B_2, \dots, B_m$$

Tal que:

$$\{B_1, B_2, \dots, B_m\} \subseteq \{A_1, A_2, \dots, A_n\}$$

Teoría de diseño de BBDD relacionales

► Dependencia Funcional trivial

◦ Ejemplos:

- Dada la relación: CONTRATADO(nroContratado, dni, nombrePersona, inicioActividad)
- Clave : {nroContratado, dni}
- Dfs:
 - df1) dni \rightarrow nombrePersona
 - df2) nroContratado, dni \rightarrow inicioActividad

Ejemplos de dependencias funcionales triviales:

dni \rightarrow dni

nroContratado, dni \rightarrow nroContratado

EJERCICIOS

Teoría de diseño de BBDD relacionales

Ejercicio

Dada la relación

EMPLEADO (nroEmpleado, nombre, fechaNac)

- Determinar si las siguientes dependencias funcionales (dfs) son correctas para la relación EMPLEADO

Enunciación	correcto/incorrecto
fechaNac \rightarrow nombre, nroEmpleado	
nroEmpleado \rightarrow fechaNac, nombre	

Donde:

- El nombre del empleado se puede repetir
- En una fecha de nacimiento, nacen muchas personas pero una persona posee una única fecha de nacimiento

Teoría de diseño de BBDD relacionales

Ejercicio

Dada la relación

EMPLEADO (nroEmpleado, nombre, fechaNac)

- Determinar si las siguientes dependencias funcionales (dfs) son correctas para la relación EMPLEADO

Enunciación	correcto/incorrecto
fechaNac \rightarrow nombre, nroEmpleado	incorrecto
nroEmpleado \rightarrow fechaNac, nombre	correcto

Donde:

- El nombre del empleado se puede repetir
- En una fecha de nacimiento, nacen muchas personas pero una persona posee una única fecha de nacimiento

Teoría de diseño de BBDD relacionales

Ejercicio

Dada la relación

VENTAS(codCliente, nombre, codVenta, monto)

- Donde:
 - Un cliente realiza muchas compras
 - Una compra es realizada por un solo cliente
 - El monto representa el valor total de la compra realizada por un cliente
- Determinar las dependencias funcionales (dfs) válidas en VENTAS

Teoría de diseño de BBDD relacionales

Ejercicio

Dada la relación

VENTAS(codCliente, nombre, codVenta, monto)

- Donde:
 - Un cliente realiza muchas compras
 - Una compra es realizada por un solo cliente
 - El monto representa el valor total de la compra realizada por un cliente
- Determinar las dependencias funcionales (dfs) válidas en VENTAS
 - df1) codCliente \rightarrow nombre
 - df2) codVenta \rightarrow codCliente, monto

Teoría de diseño de BBDD relacionales

Ejercicio

Dada la relación

VENTAS1(codCliente, nombreCliente, codVenta, monto)

- Donde:
 - Un cliente realiza muchas compras
 - Una compra puede ser realizada por mas de un cliente
 - El monto representa el valor total de la compra realizada por cada cliente
- Determinar las dependencias funcionales (dfs) válidas en VENTAS1

Teoría de diseño de BBDD relacionales

Ejercicio

Dada la relación

VENTAS1(codCliente, nombre, codVenta, monto)

- Donde:
 - Un cliente realiza muchas compras
 - Una compra puede ser realizada por mas de un cliente
 - El monto representa el valor total de la compra realizada por cada cliente
- Determinar las dependencias funcionales (dfs) válidas en VENTAS1
 - df1) codCliente \rightarrow nombre
 - df2) codVenta, codCliente \rightarrow monto

Teoría de diseño de BBDD relacionales

Ejercicio

Dada la relación

PERSONAEMPLEADA(dni, nombre, domicilio, depto, flngDepto, codEmpDepto, jefe)

Donde

- *En cada departamento hay un jefe para todos los empleados. Un mismo jefe puede estar asignado a mas de un departamento*
- *Una persona puede trabajar en mas de un departamento y en cada uno de ellos puede tener un código de empleado diferente*
- *El código de empleado no se repite para un mismo departamento, puede repetirse en diferentes departamentos*
- *Domicilio indica el lugar en el que vive una persona. Mas de una persona pueden vivir en el mismo domicilio*
- *Nombre indica la forma en la que se llama una persona. Notar que diferentes personas pueden llamarse igual*

Determinar las dependencias funcionales válidas en
PERSONAEMPLEADA

Ejercicio

Dada la relación

PERSONAEMPLEADA(dni, nombre, domicilio, codDepto, flngDepto, codEmpDepto, jefe)

Donde

- *En cada departamento hay un jefe para todos los empleados. Un mismo jefe puede estar asignado a mas de un departamento*
- *Una persona puede trabajar en mas de un departamento y en cada uno de ellos puede tener un código de empleado diferente*
- *El código de empleado no se repite para un mismo departamento, puede repetirse en diferentes departamentos*
- *Domicilio indica el lugar en el que vive una persona. Mas de una persona pueden vivir en el mismo domicilio*
- *Nombre indica la forma en la que se llama una persona. Notar que diferentes personas pueden llamarse igual*

Determinar las dependencias funcionales válidas en **PERSONAEMPLEADA**

df1) dni \rightarrow nombre, domicilio

df2) codDepto \rightarrow jefe

df3) codDepto, codEmpDepto \rightarrow dni, flngDepto

df4) codDepto, dni \rightarrow codEmpDepto , flngDepto

Teoría de diseño de BBDD relacionales

Ejercicio

Dada la relación

PERSONAEMPLEADA1(dni, nombre, domicilio, depto, flngDepto, codEmpDepto, jefe)

Donde

- *Cada persona en un departamento tiene asignado a un jefe. El mismo jefe puede estar asignado a diferentes personas de un departamento o de diversos departamentos*
- *Una persona puede trabajar en más de un departamento y en cada uno de ellos puede tener un código de empleado diferente*
- *El código de empleado no se repite para un mismo departamento, puede repetirse en diferentes departamentos*
- *Domicilio indica el lugar en el que vive una persona. Más de una persona pueden vivir en el mismo domicilio*
- *Nombre indica la forma en la que se llama una persona. Notar que diferentes personas pueden llamarse igual*

Determinar las dependencias funcionales válidas en **PERSONAEMPLEADA1**

Ejercicio

Dada la relación

PERSONAEMPLEADA1(dni, nombre, domicilio, depto, flngDepto, codEmpDepto, jefe)

Donde

- *Cada persona en un departamento tiene asignado a un jefe. El mismo jefe puede estar asignado a diferente personas de un departamento o de diversos departamentos*
- *Una persona puede trabajar en mas de un departamento y en cada uno de ellos puede tener un código de empleado diferente*
- *El código de empleado no se repite para un mismo departamento, puede repetirse en diferentes departamentos*
- *Domicilio indica el lugar en el que vive una persona. Mas de una persona pueden vivir en el mismo domicilio*
- *Nombre indica la forma en la que se llama una persona. Notar que diferentes personas pueden llamarse igual*

Determinar las dependencias funcionales válidas en **PERSONAEMPLEADA1**

df1) dni → nombre, domicilio

df2) codDepto, codEmpDepto → dni, jefe, flngDepto

df3) codDepto, dni → codEmpDepto, jefe, flngDepto

Teoría de diseño para bases de datos relaciones

CLAVE DE UNA RELACIÓN



Teoría de diseño de BBDD relacionales

► Clave de una relación

- Los atributos $\{A_1, A_2, \dots, A_n\}$ son la clave de una relación R si cumplen:
 - $\{A_1, A_2, \dots, A_n\}$ determinan funcionalmente a todos los restantes atributos de la relación R
 - No existe un subconjunto de $\{A_1, A_2, \dots, A_n\}$ que determine funcionalmente a todos los atributos de R –*Esto implica que una clave es un conjunto minimal*–

Teoría de diseño de BBDD relacionales

► Clave de una relación

◦ Ejemplo

- PERSONA(dni, nombre, edad, fechaNacimiento)
- df1: dni \rightarrow nombre, edad, fechaNac
- **Clave: {dni}**

Teoría de diseño para bases de datos relaciones

CLAVE CANDIDATA DE UNA RELACIÓN

Teoría de diseño de BBDD relacionales

► Clave de una relación /Clave candidata

- En caso de existir dos o mas conjuntos de atributos $\{A_1, A_2, \dots, A_n\}$, $\{B_1, B_2, \dots, B_k\}$, ... $\{N_1, N_2, \dots, N_m\}$ en una relación R tales que
 - $\{A_1, A_2, \dots, A_n\}$ determinan funcionalmente a todos los restantes atributos de la relación R
 - $\{B_1, B_2, \dots, B_k\}$, ... y $\{N_1, N_2, \dots, N_m\}$ también por si mismos determinan al resto de los atributos de R
 - No existe un subconjunto de $\{A_1, A_2, \dots, A_n\}$ o $\{B_1, B_2, \dots, B_k\}$, ... o $\{N_1, N_2, \dots, N_m\}$ que determine funcionalmente a todos los atributos de R

Entonces $\{A_1, A_2, \dots, A_n\}$, $\{B_1, B_2, \dots, B_k\}$, ... $\{N_1, N_2, \dots, N_m\}$ son **CLAVES CANDIDATAS** para la relación R

Teoría de diseño de BBDD relacionales

► Clave de una relación /Clave candidata

- Ejemplo:
 - Dada la relación: PERSONA(dni, nombre, edad, fechaNacimiento, nroLegajo)
 - *Donde*
 - *Una persona posee un único número de legajo asignado*
 - *Un número de legajo pertenece a una sola persona*
- Se pueden enunciar las siguientes dfs
 - df1) dni \rightarrow nombre, edad, fechaNac, nroLegajo
 - df2) nroLegajo \rightarrow nombre, edad, fechaNac, dni

Clave candidata 1 (cc1): {dni}

Clave candidata 2 (cc2): {nroLegajo}

Teoría de diseño de BBDD relacionales

▶ Clave de una relación /Clave candidata

◦ Ejemplo:

- Dada la relación: PERSONA(dni, nombre, edad, fechaNacimiento, nroLegajo, carrera)
 - *Donde*
 - *Una persona puede cursar diversas carreras*
 - *Nombre indica como se llama la persona*
 - *Una persona posee un único número de legajo asignado para cada carrera que cursa*
 - *Un número de legajo pertenece a una sola persona de una carrera*
- df1) dni -> nombre, edad, fechaNac
df2) nroLegajo, carrera -> dni
df3) dni, carrera -> nroLegajo

Clave candidata 1 (cc1): {nroLegajo, carrera }

Clave candidata 2 (cc2): {dni, carrera }

Teoría de diseño para bases de datos relaciones

SUPERCLAVE DE UNA RELACIÓN

Teoría de diseño de BBDD relacionales

► Superclave de una relación

- “Super conjunto” de una clave
- Los atributos $\{A1, A2, \dots, An\}$ son la superclave de una relación R si cumplen:
 - $\{A1, A2, \dots, An\}$ determinan funcionalmente a todos los restantes atributos de la relación R
 - Notar que:
 - Una clave esta contenida en una superclave
 - Una superclave no necesariamente es minimal (como lo es la clave por la segunda condición de su definición)

Teoría de diseño de BBDD relacionales

► Superclave de una relación

◦ Ejemplo

- PERSONA(dni, nombre, edad, fechaNacimiento)
- $df1: dni \rightarrow nombre, edad, fechaNac$
- **superclave: {dni, nombre}**

EJERCICIOS

Teoría de diseño de BBDD relacionales

Ejercicio

Dada la relación

EMPLEADO (nroEmpleado, nombre, fechaNac)

- Determinar si las siguientes dependencias funcionales (dfs) son correctas para la relación EMPLEADO
- Donde:
 - El nombre del empleado se puede repetir
 - En una fecha de nacimiento, nacen muchas personas pero una persona posee una única fecha de nacimiento
- $df1 \rightarrow nroEmpleado \rightarrow fechaNac, nombre$

Clave: {nroEmpleado}

Teoría de diseño de BBDD relacionales

Ejercicio

Dada la relación

VENTAS(codCliente, nombre, codVenta, monto)

- Donde:
 - Un cliente realiza muchas compras
 - Una compra es realizada por un solo cliente
 - El monto representa el valor total de la compra realizada por un cliente
- Determinar las dependencias funcionales (dfs) válidas en VENTAS
 - df1) codCliente \rightarrow nombre
 - df2) codVenta \rightarrow codCliente, monto

Clave: {codVenta}

Teoría de diseño de BBDD relacionales

Ejercicio

Dada la relación

VENTAS1(codCliente, nombre, codVenta, monto)

- Donde:
 - Un cliente realiza muchas compras
 - Una compra puede ser realizada por mas de un cliente
 - El monto representa el valor total de la compra realizada por cada cliente
- Determinar las dependencias funcionales (dfs) válidas en VENTAS1
 - df1) codCliente \rightarrow nombre
 - df2) codVenta, codCliente \rightarrow monto

Clave: {codVenta, codCliente}

Ejercicio

Dada la relación

PERSONAEMPLEADA(dni, nombre, domicilio, depto, flngDepto, codEmpDepto, jefe)

Donde

- *En cada departamento hay un jefe para todos los empleados. Un mismo jefe puede estar asignado a mas de un departamento*
- *Una persona puede trabajar en mas de un departamento y en cada uno de ellos puede tener un código de empleado diferente*
- *El código de empleado no se repite para un mismo departamento, puede repetirse en diferentes departamentos*
- *Domicilio indica el lugar en el que vive una persona. Mas de una persona pueden vivir en el mismo domicilio*
- *Nombre indica la forma en la que se llama una persona. Notar que diferentes personas pueden llamarse igual*

Determinar las dependencias funcionales válidas en **PERSONAEMPLEADA**

df1) dni \rightarrow nombre, domicilio

df2) depto \rightarrow jefe

df3) codDepto, codEmpDepto \rightarrow dni

df4) codDepto, dni \rightarrow codEmpDepto

cc1: {codDepto, codEmpDepto}

cc2: {codDepto, dni}

Ejercicio

Dada la relación

PERSONAEMPLEADA1(dni, nombre, domicilio, depto, flngDepto, codEmpDepto, jefe)

Donde

- *Cada persona en un departamento tiene asignado a un jefe. El mismo jefe puede estar asignado a diferente personas de un departamento o de diversos departamentos*
- *Una persona puede trabajar en mas de un departamento y en cada uno de ellos puede tener un código de empleado diferente*
- *El código de empleado no se repite para un mismo departamento, puede repetirse en diferentes departamentos*
- *Domicilio indica el lugar en el que vive una persona. Mas de una persona pueden vivir en el mismo domicilio*
- *Nombre indica la forma en la que se llama una persona. Notar que diferentes personas pueden llamarse igual*

Determinar las dependencias funcionales válidas en **PERSONAEMPLEADA1**

df1) dni \rightarrow nombre, domicilio

df2) codDepto, codEmpDepto \rightarrow dni, jefe

df3) codDepto, dni \rightarrow codEmpDepto, jefe

cc1: {codDepto, codEmpDepto}

cc2: {codDepto, dni}

Teoría de diseño para bases de datos relaciones

Axiomas de Armstrong



Teoría de diseño de BBDD relacionales

¿Cómo deducir nuevas
dependencias funcionales
a partir de un conjunto
dado?

Teoría de diseño de BBDD relacionales

▶ Axiomas de Armstrong

- Permiten inferir nuevas dependencias funcionales dado un conjunto base que resultó evidente
- Aplicándolos halla un conjunto completo y seguro donde todas las dependencias funcionales halladas son correctas
- Al generar todas las dependencias funcionales algunas son triviales

Teoría de diseño de BBDD relacionales

▶ Axiomas de Armstrong

- Axiomas Básicos
 - Reflexión
 - Aumento
 - Transitividad
- Axiomas que se deducen a partir de los básicos
 - Unión
 - Descomposición
 - Pseudotransitividad

Teoría de diseño de BBDD relacionales

► Axiomas de Armstrong

► Reflexión:

X es un conjunto de atributos y $Y \subseteq X$ entonces $X \twoheadrightarrow Y$

Sabemos que $a \twoheadrightarrow a$, luego se puede decir que $a, b \twoheadrightarrow a$

Demostración:

Si $Y \subseteq X$ y existen dos tuplas diferentes de R tales que
 $t1[x] = t2[x]$ por definición de dependencia funcional
 $t1[y] = t2[y]$

Teoría de diseño de BBDD relacionales

► Axiomas de Armstrong

➤ Aumento

Si $X \rightarrow Y$; Z es un conjunto de atributos, entonces
 $Z, X \rightarrow Z, Y$

Demostración:

Asumamos que $X \rightarrow Y$ vale pero $X, Z \rightarrow Y, Z$ no vale
Si $X \rightarrow Y$ entonces cada vez que

1) $t1[x] = t2[x]$ implica

2) $t1[y] = t2[y]$

Por otro lado, cada vez que

3) $t1[x, z] = t2[x, z]$ implica

4) $t1[y, z] \neq t2[y, z]$

De 1) y 3) se deduce $t1[z] = t2[z]$

De 2) y 4) se deduce que $t1[y, z] = t2[y, z]$

Teoría de diseño de BBDD relacionales

▶ Axiomas de Armstrong

➤ Transitividad

Si $X \twoheadrightarrow Y$; $Y \twoheadrightarrow Z$, entonces $X \twoheadrightarrow Z$

Demostración:

1) $X \twoheadrightarrow Y$

2) $Y \twoheadrightarrow Z$

$t1[x]=t2[x]$ implica por 1)

$t1[y]=t2[y]$ implica por 2)

$t1[z]=t2[z]$ entonces

$X \twoheadrightarrow Z$

Teoría de diseño de BBDD relacionales

► Axiomas de Armstrong

► Unión

Si $X \twoheadrightarrow Y$; $X \twoheadrightarrow Z$, entonces $X \twoheadrightarrow Y, Z$

Demostración:

1) $X \twoheadrightarrow Y$

2) $X \twoheadrightarrow Z$

Si $X \twoheadrightarrow Y$, por aumentación vale que $X \twoheadrightarrow XY$

Si $X \twoheadrightarrow Z$, por aumentación vale que $X, Y \twoheadrightarrow Y, Z$

Luego por transitividad, $X \twoheadrightarrow Y, Z$

Teoría de diseño de BBDD relacionales

▶ Axiomas de Armstrong

▶ Descomposición

Si $X \twoheadrightarrow Y, Z$, entonces $X \twoheadrightarrow Y$, $X \twoheadrightarrow Z$

Demostración:

$X \twoheadrightarrow Y, Z$

por reflexividad vale que $Y, Z \twoheadrightarrow Y$

Luego, por transitividad $X \twoheadrightarrow Y$

Por reflexividad también vale que $Y, Z \twoheadrightarrow Z$

Luego por transitividad, también vale que $X \twoheadrightarrow Z$

Teoría de diseño de BBDD relacionales

▶ Axiomas de Armstrong

➤ Pseudotransitividad

Si $X \twoheadrightarrow Y$; $Y, Z \twoheadrightarrow W$ entonces $X, Z \twoheadrightarrow W$

Demostración:

$X \twoheadrightarrow Y$

por aumento vale que $X, Z \twoheadrightarrow Y, Z$

Por otro lado se sabe que $Y, Z \twoheadrightarrow W$

Luego por transitividad, vale que $X, Z \twoheadrightarrow W$

Teoría de diseño para bases de datos relaciones

Clausura de un conjunto de ATRIBUTOS



Teoría de diseño de BBDD relacionales

► Clave de una relación

- Los atributos $\{A_1, A_2, \dots, A_n\}$ son la clave de una relación R si cumplen:
 - $\{A_1, A_2, \dots, A_n\}$ determinan funcionalmente a todos los restantes atributos de la relación R
 - No existe un subconjunto de $\{A_1, A_2, \dots, A_n\}$ que determine funcionalmente a todos los atributos de R –*Esto implica que una clave es un conjunto minimal*–

Teoría de diseño de BBDD relacionales

¿Cómo se puede comprobar que un conjunto de atributos $\{A1, A2, \dots, An\}$ es la clave de una relación?

Teoría de diseño de BBDD relacionales

► Clausura de un conjunto de atributos (X^+)

Sea F un conjunto de dependencias funcionales sobre un esquema R y sea X un subconjunto de R .

La clausura de X respecto de F , se denota X^+ y es el conjunto de atributos A tal que la dependencia $X \rightarrow A$ puede deducirse a partir de F , por los axiomas de Armstrong

Es decir, X^+ son todos los atributos determinados por X en R

Teoría de diseño de BBDD relacionales

► Clausura de un conjunto de atributos (X^+)

Algoritmo para encontrar X^+

Result := X

While (hay cambios en result) do

For (cada dependencia funcional $Y \rightarrow Z$ en F) do

if ($Y \subseteq \text{result}$) then

result := result \cup Z

Teoría de diseño de BBDD relacionales

Continuará la próxima clase...



Bibliografía de la clase

- Garcia-Molina, H. (2008). *Database systems: the complete book*. Pearson Education India.
- Ullman, J. D. (1988). Principles of database and knowledge-base systems.