

Bases de Datos

Dependencias Funcionales y Anomalías

Lic. Andy Ledesma García
Lic. Víctor M. Cardentey Fundora
Dra. C. Lucina García Hernández

Departamento de Computación
Facultad de Matemática y Computación
Universidad de La Habana

Licenciatura en Ciencia de Datos

13 de febrero de 2024

Lo que usted aprenderá

1. Reconocer las restricciones de integridad en un escenario y representarlas adecuadamente en una base de datos relacional.

Lo que usted aprenderá

1. Reconocer las restricciones de integridad en un escenario y representarlas adecuadamente en una base de datos relacional.
2. Detectar anomalías en una base de datos relacional.



¿Qué es una llave candidata?

¿Qué es una llave candidata?

Llave candidata

Un conjunto de uno o más atributos $K = \{A_1, A_2, \dots, A_n\}$ es una llave candidata de la relación R si cumple las siguiente propiedades:

1. **Unicidad:** En cualquier momento dado, no existen dos tuplas distintas de R con los mismos valores para A_1, A_2, \dots, A_n .
2. **Minimalidad:** Ningún subconjunto propio de K tiene la propiedad de unicidad.

¿Cómo encontrar las llaves candidatas de una relación?

La idea es muy fácil

Sea $U = \{A_1, \dots, A_n\}$ el conjunto universo de los atributos de una relación R , por cada subconjunto de atributos $X \subseteq U$ comprobar la unicidad y minimalidad.

Un conjunto C es minimal con respecto a una propiedad P si y sólo si:

1. $P(C)$
2. $\nexists C' \subset C : P(C')$

Un conjunto C es minimal con respecto a una propiedad P si y sólo si:

1. $P(C)$
2. $\nexists C' \subset C : P(C')$

Para comprobar la minimalidad debemos comprobar la unicidad

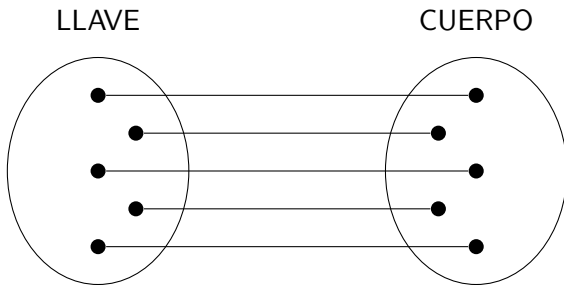
¿Qué es la unicidad?

Unicidad

Sea $K = \{A_1, A_2, \dots, A_n\}$ un conjunto de atributos de una relación R , en cualquier momento dado, no existen dos tuplas distintas de R con los mismos valores para A_1, A_2, \dots, A_n .

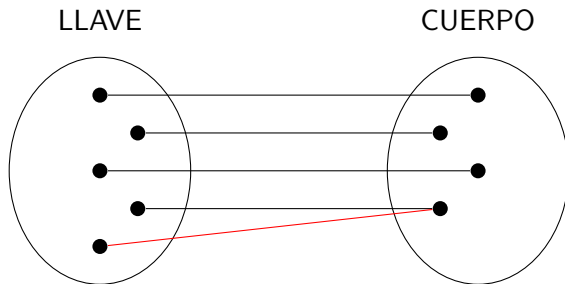
Unicidad

Sea $K = \{A_1, A_2, \dots, A_n\}$ un conjunto de atributos de una relación R , en cualquier momento dado, no existen dos tuplas distintas de R con los mismos valores para A_1, A_2, \dots, A_n .



Unicidad

Sea $K = \{A_1, A_2, \dots, A_n\}$ un conjunto de atributos de una relación R , en cualquier momento dado, no existen dos tuplas distintas de R con los mismos valores para A_1, A_2, \dots, A_n .



No necesariamente es una función inyectiva

¿Esta definición ayuda a obtener un algoritmo?

Determinar la existencia de una función entre dos conjuntos de elementos desconocidos

¿Esta definición ayuda a obtener un algoritmo?

Determinar la existencia de una función entre dos conjuntos de elementos desconocidos

¿Cómo podemos hacer esto?

Ya sabemos que existen algunas

- ▶ Los proveedores de correo (Google, Yahoo, Outlook, etc.) asocian al correo algunos datos personales como el nombre y apellido.
- ▶ Existe un convenio internacional en el que los países acuerdan códigos para identificar la ubicación geográfica (código postal).

Ya sabemos que existen algunas

- ▶ Los proveedores de correo (Google, Yahoo, Outlook, etc.) asocian al correo algunos datos personales como el nombre y apellido.
- ▶ Existe un convenio internacional en el que los países acuerdan códigos para identificar la ubicación geográfica (código postal).

Si se desea desarrollar un sistema que requiera tanto los datos personales como la ubicación geográfica del usuario, cuál sería la llave primaria de la relación Usuario.

Buscando una llave candidata

Usuario(Email, Nombre, Apellido, C. Postal, Provincia, País)

Buscando una llave candidata

Usuario(Email, Nombre, Apellido, C. Postal, Provincia, País)

- ▶ Si conocemos el email de un usuario también conocemos su nombre y apellido.

Email → Nombre, Apellido

Buscando una llave candidata

Usuario(Email, Nombre, Apellido, C. Postal, Provincia, País)

- ▶ Si conocemos el email de un usuario también conocemos su nombre y apellido.

Email → Nombre, Apellido

- ▶ Si conocemos el código postal de un usuario también conocemos su provincia y país.

C. Postal → Provincia, País

Buscando una llave candidata

Usuario(Email, Nombre, Apellido, C. Postal, Provincia, País)

- ▶ Si conocemos el email de un usuario también conocemos su nombre y apellido.

Email → Nombre, Apellido

- ▶ Si conocemos el código postal de un usuario también conocemos su provincia y país.

C. Postal → Provincia, País

¿Cuál sería una llave candidata de esta relación?

¿Y si componemos las funciones que ya conocemos?

Email, C. Postal \rightarrow Nombre, Apellido, Provincia, País

¿Y si componemos las funciones que ya conocemos?

Email, C. Postal \rightarrow Nombre, Apellido, Provincia, País

- ▶ Supongamos un usuario con email luis.fonseca@gmail.com

luis.fonseca@gmail.com \rightarrow Luis, Fonseca

- ▶ Supongamos que tiene el código zip 10400

10400 \rightarrow La Habana, Cuba

¿Y si componemos las funciones que ya conocemos?

Email, C. Postal \rightarrow Nombre, Apellido, Provincia, País

- Supongamos un usuario con email luis.fonseca@gmail.com

luis.fonseca@gmail.com \rightarrow Luis, Fonseca

- Supongamos que tiene el código zip 10400

10400 \rightarrow La Habana, Cuba

luis.fonseca@gmail.com, 10400 \rightarrow Luis, Fonseca, La Habana, Cuba

Dependencia Funcional

Dada una relación R y los atributos X, Y de R , se dice que Y depende funcionalmente de X si y sólo si el valor de X en cada tupla de R determina el valor de Y en dicha tupla. Se representa como $R.X \rightarrow R.Y$ o simplemente

$$X \rightarrow Y$$

Notación

- ▶ Atributo simple: A, B, C, D, E
- ▶ Atributo compuesto (conjunto de atributos simples): W, X, Y, Z

Cumplimiento de una Dependencia Funcional

Cumplimiento de una Dependencia Funcional

Función f

$$x_1 = x_2 \implies f(x_1) = f(x_2)$$

Cumplimiento de una Dependencia Funcional

Función f

$$x_1 = x_2 \implies f(x_1) = f(x_2)$$

Dependencia Funcional $X \rightarrow Y$

$$t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

Cumplimiento de una Dependencia Funcional

Función f

$$x_1 = x_2 \implies f(x_1) = f(x_2)$$

Dependencia Funcional $X \rightarrow Y$

$$t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

<u>#E</u>	Grupo	Provincia
1	211	Pinar del Río
2	212	La Habana
3	213	La Habana
15	211	Pinar del Río

Cumplimiento de una Dependencia Funcional

Función f

$$x_1 = x_2 \implies f(x_1) = f(x_2)$$

Dependencia Funcional $X \rightarrow Y$

$$t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

<u>#E</u>	Grupo	Provincia
1	211	Pinar del Río
2	212	La Habana
3	213	La Habana
15	211	Pinar del Río

► $\#E \rightarrow \text{Grupo, Provincia}$

Cumplimiento de una Dependencia Funcional

Función f

$$x_1 = x_2 \implies f(x_1) = f(x_2)$$

Dependencia Funcional $X \rightarrow Y$

$$t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

<u>#E</u>	Grupo	Provincia
1	211	Pinar del Río
2	212	La Habana
3	213	La Habana
15	211	Pinar del Río

- ▶ $\#E \rightarrow \text{Grupo}, \text{Provincia}$
- ▶ $\text{Grupo} \rightarrow \text{Provincia}$

Cumplimiento de una Dependencia Funcional

Función f

$$x_1 = x_2 \implies f(x_1) = f(x_2)$$

<u>#E</u>	Grupo	Provincia
1	211	Pinar del Río
2	212	La Habana
3	213	La Habana
15	211	Pinar del Río

Dependencia Funcional $X \rightarrow Y$

$$t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

- ▶ $\#E \rightarrow \text{Grupo}, \text{Provincia}$
- ▶ $\text{Grupo} \rightarrow \text{Provincia}$

¿Provincia \rightarrow Grupo?

Cumplimiento de una Dependencia Funcional

Función f

$$x_1 = x_2 \implies f(x_1) = f(x_2)$$

<u>#E</u>	Grupo	Provincia
1	211	Pinar del Río
2	212	La Habana
3	213	La Habana
15	211	Pinar del Río

Dependencia Funcional $X \rightarrow Y$

$$t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$$

- ▶ $\#E \rightarrow \text{Grupo}, \text{Provincia}$
- ▶ $\text{Grupo} \rightarrow \text{Provincia}$

¿Provincia \rightarrow Grupo?

¿Cómo representar una relación?

Esquema relacional

Un **esquema relacional** expresado por $R(U, F)$ constituye una manera abreviada de representar la descripción de una relación mediante:

- ▶ Su nombre R .
- ▶ El conjunto de atributos que la componen U , denominado universo de la relación.
- ▶ El conjunto de dependencias funcionales F que se cumple en R .

Utilidad de las dependencias funcionales

- Especificar las restricciones en el conjunto de instancias legales de una relación R (instancias r de R que satisfacen un conjunto de dependencias funcionales F):

F se cumple en R

Utilidad de las dependencias funcionales

- Especificar las restricciones en el conjunto de instancias legales de una relación R (instancias r de R que satisfacen un conjunto de dependencias funcionales F):

F se cumple en R

- Probar si una instancia r de una relación R es legal bajo un conjunto de dependencias funcionales F :

r satisface a F

Utilidad de las dependencias funcionales

- Especificar las restricciones en el conjunto de instancias legales de una relación R (instancias r de R que satisfacen un conjunto de dependencias funcionales F):

F se cumple en R

- Probar si una instancia r de una relación R es legal bajo un conjunto de dependencias funcionales F :

r satisface a F

- ¿Y los algoritmos?

AUTOMATE



ALL THE THINGS

Ejemplo de aplicación del algoritmo para comprobar la unicidad

Sea $R(U, F)$ con:

► $U = \{A, B, C, D, E, G\}$

► $F = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow EG, BE \rightarrow C\}$

comprobar si CD cumple la unicidad.

Ejemplo de aplicación del algoritmo para comprobar la unicidad

Sea $R(U, F)$ con:

► $U = \{A, B, C, D, E, G\}$

► $F = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow EG, BE \rightarrow C\}$

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

Ejemplo de aplicación del algoritmo para comprobar la unicidad

Sea $R(U, F)$ con:

► $U = \{A, B, C, D, E, G\}$

► $F = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow EG, BE \rightarrow C\}$

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

Ejemplo de aplicación del algoritmo para comprobar la unicidad

Sea $R(U, F)$ con:

► $U = \{A, B, C, D, E, G\}$

► $F = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow EG, BE \rightarrow C\}$

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

1. $X_1 = ACD$

Ejemplo de aplicación del algoritmo para comprobar la unicidad

Sea $R(U, F)$ con:

► $U = \{A, B, C, D, E, G\}$

► $F = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, \textcolor{red}{ACD} \rightarrow B, D \rightarrow EG, BE \rightarrow C\}$

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

1. $X_1 = \textcolor{red}{ACD}$

Ejemplo de aplicación del algoritmo para comprobar la unicidad

Sea $R(U, F)$ con:

► $U = \{A, B, C, D, E, G\}$

► $F = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow EG, BE \rightarrow C\}$

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

1. $X_1 = ACD$

2. $X_2 = ABCD$

Ejemplo de aplicación del algoritmo para comprobar la unicidad

Sea $R(U, F)$ con:

► $U = \{A, B, C, D, E, G\}$

► $F = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, \textcolor{red}{D} \rightarrow EG, BE \rightarrow C\}$

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

1. $X_1 = ACD$

2. $X_2 = ABC\textcolor{red}{D}$

Ejemplo de aplicación del algoritmo para comprobar la unicidad

Sea $R(U, F)$ con:

- ▶ $U = \{A, B, C, D, E, G\}$
- ▶ $F = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow EG, BE \rightarrow C\}$

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

1. $X_1 = ACD$
2. $X_2 = ABCD$
3. $X_3 = ABCDEG$

Ejemplo de aplicación del algoritmo para comprobar la unicidad

Sea $R(U, F)$ con:

- ▶ $U = \{A, B, C, D, E, G\}$
- ▶ $F = \{AB \rightarrow C, C \rightarrow A, BC \rightarrow D, ACD \rightarrow B, D \rightarrow EG, BE \rightarrow C\}$

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

1. $X_1 = ACD$
2. $X_2 = ABCD$
3. $X_3 = ABCDEG$

$X_3 = U$, se cumple la unicidad

Implicación lógica

Sea un esquema relacional $R(U, F)$ y $X \rightarrow Y$ una dependencia funcional. Se dice que F implica lógicamente a $X \rightarrow Y$ o que $X \rightarrow Y$ se deduce lógicamente de F si cada instancia r de R que satisfaga las dependencias funcionales en F también satisface $X \rightarrow Y$.

$$F \models X \rightarrow Y$$

Clausura de un conjunto de atributos

Sea un esquema relacional $R(U, F)$, un conjunto de atributos X tal que $X \subseteq U$. La clausura del conjunto de atributos X con respecto a un conjunto de dependencias funcionales F – denotada por X_F^+ o abreviadamente X^+ – es el conjunto de atributos simples que se determina funcionalmente por X a partir de las dependencias funcionales de F .

$$X_F^+ = \{A_i \in U \mid F \models X \rightarrow A_i\}$$

Mejorando la definición de llave candidata

Sea K un conjunto de atributos $\{A_1, A_2, \dots, A_n\}$ de una esquema relacional $R(U, F)$ es llave candidata del esquema si cumple las siguientes propiedades:

1. **Unicidad:** $K_F^+ = U$
2. **Minimalidad:** Ningún subconjunto propio de K tiene la propiedad de unicidad.

Mejorando el algoritmo para comprobar la unicidad

El conjunto de atributos X cumple la unicidad en $R(U, F)$ si y sólo si $X_F^+ = U$

¿Qué herramientas podemos aplicar para inferir lógicamente?

¿Qué herramientas podemos aplicar para inferir lógicamente?

Axiomas de Inferencia (Armstrong, 1974)

Sea un esquema relacional $R(U, F)$:

1. **Reflexividad**: Si $Y \subseteq X \subseteq U$, entonces $X \rightarrow Y$ se deduce lógicamente de F .
2. **Aumentatividad**: Si se cumple que $X \rightarrow Y$ y $Z \subseteq U$, entonces $XZ \rightarrow YZ$.
3. **Transitividad**: Si se cumple que $X \rightarrow Y$ y $Y \rightarrow Z$, entonces $X \rightarrow Z$.

Armstrong, W. W. [1974]. *"Dependency structures of data base relationships,"*

Proc. 1974 IFIP Congress, pp. 580-583, North Holland, Amsterdam.

Lemas derivados

Sea un esquema relacional $R(U, F)$:

1. **Composición:** $\{X \rightarrow Y, W \rightarrow Z \mid W \subseteq X\} \models X \rightarrow YZ$
2. **Descomposición:** $\{X \rightarrow Y, Z \subseteq Y\} \models X \rightarrow Z$
3. **Pseudotransitividad:** $\{X \rightarrow Y, WY \rightarrow Z\} \models XW \rightarrow Z$

Clausura de un conjunto de dependencias funcionales

Sea un esquema relacional $R(U, F)$. La clausura del conjunto de dependencias F – denotada por F^+ – es el conjunto de las dependencias funcionales implicadas lógicamente por F . Formalmente, se define como:

$$F^+ = \{X \rightarrow Y \mid F \models X \rightarrow Y\}$$

Trivialidad de una dependencia funcional

Sean X y Y atributos no necesariamente simples, tal que $X \cap Y = \emptyset$.

Trivialidad de una dependencia funcional

Sean X y Y atributos no necesariamente simples, tal que $X \cap Y = \emptyset$.

$$X \rightarrow X$$

Trivial

Trivialidad de una dependencia funcional

Sean X y Y atributos no necesariamente simples, tal que $X \cap Y = \emptyset$.

$X \rightarrow X$
Trivial

$X \rightarrow XY$
No trivial

Trivialidad de una dependencia funcional

Sean X y Y atributos no necesariamente simples, tal que $X \cap Y = \emptyset$.

$X \rightarrow X$
Trivial

$X \rightarrow XY$
No trivial

$X \rightarrow Y$
Completamente no trivial

¿Es viable computar F^+ ?

¿Cuántas dependencias funcionales triviales existen en F^+ ?

¿Es viable computar F^+ ?

¿Cuántas dependencias funcionales triviales existen en F^+ ?

R/ La cantidad de DF's triviales es **exponencial** con respecto a la cantidad de atributos en U .

¿Es viable computar F^+ ?

¿Cuántas dependencias funcionales triviales existen en F^+ ?

R/ La cantidad de DF's triviales es **exponencial** con respecto a la cantidad de atributos en U .

¿Cómo podemos determinar si una dependencia funcional $X \rightarrow Y$ pertenece a F^+ ?

Alternativa: Utilizar la clausura de un conjunto de atributos

Alternativa: Utilizar la clausura de un conjunto de atributos

La clausura de un conjunto de atributos es $X_F^+ = \{A_i \in U \mid F \models X \rightarrow A_i\}$

¿ $X \rightarrow Y$ pertenece a F^+ ?

1. Calcular X_F^+
2. Comprobar que $Y \subseteq X_F^+$

Alternativa: Utilizar la clausura de un conjunto de atributos

La clausura de un conjunto de atributos es $X_F^+ = \{A_i \in U \mid F \models X \rightarrow A_i\}$

¿ $X \rightarrow Y$ pertenece a F^+ ?

1. Calcular X_F^+
2. Comprobar que $Y \subseteq X_F^+$

Get X_F^+

```
1:  $S = \{X\}$ 
2:  $S' = S$ 
3: for each  $W \rightarrow Z$  in  $F$  do
4:   if  $W \subseteq S$  then
5:      $S = S \cup Z$ 
6:   end if
7: end for
8: if  $S \neq S'$  then
9:   goto line 2
10: end if
11: return  $S$ 
```

Equivalencia de conjuntos de dependencias funcionales

$$F \equiv G \Leftrightarrow F^+ = G^+$$

- ▶ Se debe considerar cada $X \rightarrow Y$ en F y determinar si X_G^+ contiene a Y .
- ▶ Se debe considerar cada $Z \rightarrow W$ en G y determinar si Z_F^+ contiene a W .

Si tenemos el conjunto de DFs del universo de atributos

¿Podemos garantizar la calidad de la base de datos?

DATABASE ISSUES



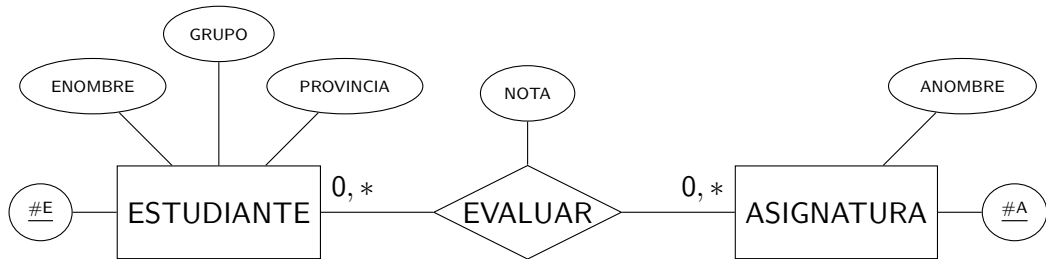
La situación

Se desea desarrollar una base de datos para registrar las notas de los estudiantes de la carrera en cada una de las asignaturas que cursan:

- ▶ De cada estudiante se conoce su identificador, su nombre, su grupo y su provincia de residencia.
- ▶ De cada asignatura se conoce su identificador y su nombre.
- ▶ Por cada asignatura se conoce la nota que obtuvo el estudiante en la evaluación final.

Además, se conoce que los estudiantes son organizados en los grupos de acuerdo a su provincia.

Primero lo primero



Metodología para obtener un esquema relacional correcto

1. Identificar el universo U de atributos del fenómeno.
2. Identificar el conjunto F de las dependencias funcionales que se establecen entre los atributos.
3. Definir el esquema relacional $R(U, F)$.

Ejemplo

1. $U = \{\#E, ENombre, Grupo, Provincia, \#A, ANombre, Nota\}$

Ejemplo

1. $U = \{\#E, ENombre, Grupo, Provincia, \#A, ANombre, Nota\}$

¿Cómo podemos obtener F a partir del diseño conceptual?

Algoritmo de extracción de dependencias funcionales

Algoritmo de extracción de dependencias funcionales

1. Por cada conjunto de entidades con un conjunto de atributos $X \subseteq U$, se añade la dependencia funcional $K \rightarrow X$ donde K es la llave del conjunto de entidades.

Algoritmo de extracción de dependencias funcionales

1. Por cada conjunto de entidades con un conjunto de atributos $X \subseteq U$, se añade la dependencia funcional $K \rightarrow X$ donde K es la llave del conjunto de entidades.
2. Por cada conjunto de interrelaciones se toma su llave K y se añade la dependencia funcional $K \rightarrow K$. Además, por cada conjunto de entidades en un extremo de cardinalidad máxima 1 en la interrelación, se añade la dependencia funcional $K - K_E \rightarrow K_E$ donde K_E es la llave del conjunto de entidades.

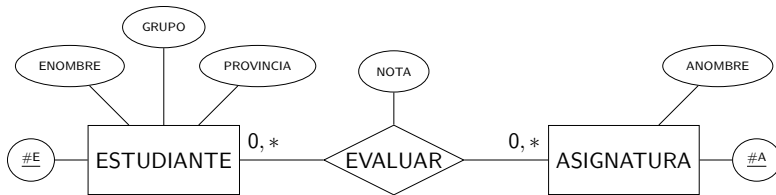
Algoritmo de extracción de dependencias funcionales

1. Por cada conjunto de entidades con un conjunto de atributos $X \subseteq U$, se añade la dependencia funcional $K \rightarrow X$ donde K es la llave del conjunto de entidades.
2. Por cada conjunto de interrelaciones se toma su llave K y se añade la dependencia funcional $K \rightarrow K$. Además, por cada conjunto de entidades en un extremo de cardinalidad máxima 1 en la interrelación, se añade la dependencia funcional $K - K_E \rightarrow K_E$ donde K_E es la llave del conjunto de entidades.
3. Por cada agregación con un conjunto de atributos $X \subseteq U$ se añade la dependencia funcional $K \rightarrow X$ donde K es la llave del conjunto de interrelaciones que encierra la agregación.

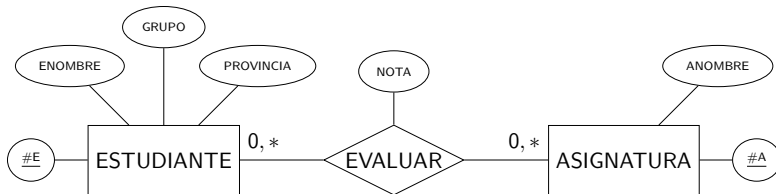
Algoritmo de extracción de dependencias funcionales

1. Por cada conjunto de entidades con un conjunto de atributos $X \subseteq U$, se añade la dependencia funcional $K \rightarrow X$ donde K es la llave del conjunto de entidades.
2. Por cada conjunto de interrelaciones se toma su llave K y se añade la dependencia funcional $K \rightarrow K$. Además, por cada conjunto de entidades en un extremo de cardinalidad máxima 1 en la interrelación, se añade la dependencia funcional $K - K_E \rightarrow K_E$ donde K_E es la llave del conjunto de entidades.
3. Por cada agregación con un conjunto de atributos $X \subseteq U$ se añade la dependencia funcional $K \rightarrow X$ donde K es la llave del conjunto de interrelaciones que encierra la agregación.
4. Añadir aquellas dependencias funcionales asociadas a otras restricciones del negocio especificadas en los requerimientos.

Ejecutando el algoritmo



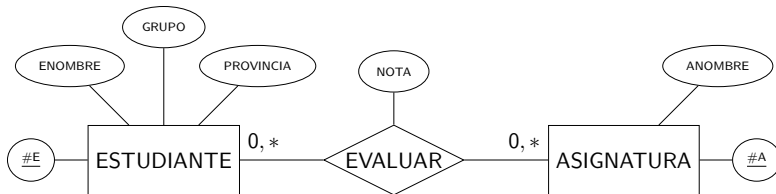
Ejecutando el algoritmo



1. Se tienen los conjuntos de entidades ESTUDIANTE y ASIGNATURA:

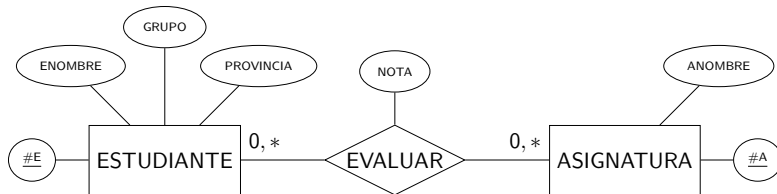
- ▶ $\#E \rightarrow \text{ENombre, Grupo, Provincia}$
- ▶ $\#A \rightarrow \text{ANombre}$

Ejecutando el algoritmo



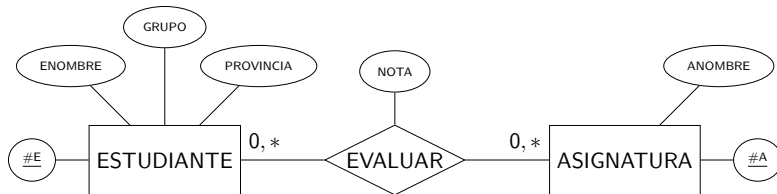
1. Se tienen los conjuntos de entidades ESTUDIANTE y ASIGNATURA:
 - ▶ $\#E \rightarrow \text{ENombre, Grupo, Provincia}$
 - ▶ $\#A \rightarrow \text{ANombre}$
2. Se tiene el conjunto de interrelaciones EVALUAR:
 - ▶ $\#E, \#A \rightarrow \#E, \#A$

Ejecutando el algoritmo



1. Se tienen los conjuntos de entidades ESTUDIANTE y ASIGNATURA:
 - ▶ $\#E \rightarrow \text{ENombre, Grupo, Provincia}$
 - ▶ $\#A \rightarrow \text{ANombre}$
2. Se tiene el conjunto de interrelaciones EVALUAR:
 - ▶ $\#E, \#A \rightarrow \#E, \#A$
3. Se tiene la agregación ASIGNATURA-EVALUADA
 - ▶ $\#E, \#A \rightarrow \text{Nota}$

Ejecutando el algoritmo



1. Se tienen los conjuntos de entidades **ESTUDIANTE** y **ASIGNATURA**:
 - ▶ $\#E \rightarrow \text{ENombre, Grupo, Provincia}$
 - ▶ $\#A \rightarrow \text{ANombre}$
2. Se tiene el conjunto de interrelaciones **EVALUAR**:
 - ▶ $\#E, \#A \rightarrow \#E, \#A$
3. Se tiene la agregación **ASIGNATURA-EVALUADA**
 - ▶ $\#E, \#A \rightarrow \text{Nota}$
4. Añadimos las restricciones planteadas en la especificación:
 - ▶ $\text{Provincia} \rightarrow \text{Grupo}$

Continuemos con el ejemplo

1. $U = \{\#E, ENombre, Grupo, Provincia, \#A, ANombre, Nota\}$
2. $F = \{$
 - $\#E \rightarrow ENombre, Grupo, Provincia$
 - $\#A \rightarrow ANombre$
 - $\#E, \#A \rightarrow \#E, \#A$
 - $\#E, \#A \rightarrow Nota$
 - $Provincia \rightarrow Grupo$ $\}$
3. Definimos el esquema relacional **Evaluaciones**(U, F) con llave $\#E, \#A$

¿Es este un buen diseño? (Anomalía de inserción)

<u>#E</u>	ENombre	Grupo	Provincia	<u>#A</u>	ANombre	Nota
e ₁	Juan	111	La Habana	a ₁	Análisis	3
e ₁	Juan	111	La Habana	a ₂	Lógica	2
e ₁	Juan	111	La Habana	a ₃	Álgebra	4
e ₁	Juan	111	La Habana	a ₄	Programación	5
e ₃	Pedro	111	La Habana	a ₃	Álgebra	4
e ₂	María	112	Matanzas	a ₁	Análisis	3
e ₂	María	112	Matanzas	a ₂	Lógica	3
e ₄	Rita	112	Mayabeque	a ₂	Lógica	3
e ₄	Rita	112	Mayabeque	a ₄	Programación	4
e ₅	Carlos	113	Pinar del Río	a ₃	Álgebra	3

¿Es este un buen diseño? (Anomalía de inserción)

<u>#E</u>	ENombre	Grupo	Provincia	<u>#A</u>	ANombre	Nota
e ₁	Juan	111	La Habana	a ₁	Análisis	3
e ₁	Juan	111	La Habana	a ₂	Lógica	2
e ₁	Juan	111	La Habana	a ₃	Álgebra	4
e ₁	Juan	111	La Habana	a ₄	Programación	5
e ₃	Pedro	111	La Habana	a ₃	Álgebra	4
e ₂	María	112	Matanzas	a ₁	Análisis	3
e ₂	María	112	Matanzas	a ₂	Lógica	3
e ₄	Rita	112	Mayabeque	a ₂	Lógica	3
e ₄	Rita	112	Mayabeque	a ₄	Programación	4
e ₅	Carlos	113	Pinar del Río	a ₃	Álgebra	3

¿Se pudiera insertar un alumno que todavía no ha recibido evaluaciones?

e₆ Marcos 111 La Habana NULL NULL NULL

¿Es este un buen diseño? (Anomalía de inserción)

<u>#E</u>	ENombre	Grupo	Provincia	<u>#A</u>	ANombre	Nota
e ₁	Juan	111	La Habana	a ₁	Análisis	3
e ₁	Juan	111	La Habana	a ₂	Lógica	2
e ₁	Juan	111	La Habana	a ₃	Álgebra	4
e ₁	Juan	111	La Habana	a ₄	Programación	5
e ₃	Pedro	111	La Habana	a ₃	Álgebra	4
e ₂	María	112	Matanzas	a ₁	Análisis	3
e ₂	María	112	Matanzas	a ₂	Lógica	3
e ₄	Rita	112	Mayabeque	a ₂	Lógica	3
e ₄	Rita	112	Mayabeque	a ₄	Programación	4
e ₅	Carlos	113	Pinar del Río	a ₃	Álgebra	3

¿Se pudiera insertar un alumno que todavía no ha recibido evaluaciones?

e₆ Marcos 111 La Habana NULL NULL NULL

¿Es este un buen diseño? (Anomalía de eliminación)

<u>#E</u>	ENombre	Grupo	Provincia	<u>#A</u>	ANombre	Nota
e_1	Juan	111	La Habana	a_1	Análisis	3
e_1	Juan	111	La Habana	a_2	Lógica	2
e_1	Juan	111	La Habana	a_3	Álgebra	4
e_1	Juan	111	La Habana	a_4	Programación	5
e_3	Pedro	111	La Habana	a_3	Álgebra	4
e_2	María	112	Matanzas	a_1	Análisis	3
e_2	María	112	Matanzas	a_2	Lógica	3
e_4	Rita	112	Mayabeque	a_2	Lógica	3
e_4	Rita	112	Mayabeque	a_4	Programación	4
e_5	Carlos	113	Pinar del Río	a_3	Álgebra	3

¿Qué ocurre si se eliminan las notas del estudiante e_5 ?

¿Es este un buen diseño? (Anomalía de eliminación)

<u>#E</u>	ENombre	Grupo	Provincia	<u>#A</u>	ANombre	Nota
e ₁	Juan	111	La Habana	a ₁	Análisis	3
e ₁	Juan	111	La Habana	a ₂	Lógica	2
e ₁	Juan	111	La Habana	a ₃	Álgebra	4
e ₁	Juan	111	La Habana	a ₄	Programación	5
e ₃	Pedro	111	La Habana	a ₃	Álgebra	4
e ₂	María	112	Matanzas	a ₁	Análisis	3
e ₂	María	112	Matanzas	a ₂	Lógica	3
e ₄	Rita	112	Mayabeque	a ₂	Lógica	3
e ₄	Rita	112	Mayabeque	a ₄	Programación	4

Se pierde la información relacionada con la provincia Pinar del Río y el grupo C113

¿Es este un buen diseño? (Anomalía de modificación)

<u>#E</u>	ENombre	Grupo	Provincia	<u>#A</u>	ANombre	Nota
e ₁	Juan	111	La Habana	a ₁	Análisis	3
e ₁	Juan	111	La Habana	a ₂	Lógica	2
e ₁	Juan	111	La Habana	a ₃	Álgebra	4
e ₁	Juan	111	La Habana	a ₄	Programación	5
e ₃	Pedro	111	La Habana	a ₃	Álgebra	4
e ₂	María	112	Matanzas	a ₁	Análisis	3
e ₂	María	112	Matanzas	a ₂	Lógica	3
e ₄	Rita	112	Mayabeque	a ₂	Lógica	3
e ₄	Rita	112	Mayabeque	a ₄	Programación	4
e ₅	Carlos	113	Pinar del Río	a ₃	Álgebra	3

¿Qué tendríamos que hacer si queremos cambiar la provincia de Juan?

¿Es este un buen diseño? (Anomalía de modificación)

<u>#E</u>	ENombre	Grupo	Provincia	<u>#A</u>	ANombre	Nota
e ₁	Juan	111	La Habana	a ₁	Análisis	3
e ₁	Juan	111	La Habana	a ₂	Lógica	2
e ₁	Juan	111	La Habana	a ₃	Álgebra	4
e ₁	Juan	111	La Habana	a ₄	Programación	5
e ₃	Pedro	111	La Habana	a ₃	Álgebra	4
e ₂	María	112	Matanzas	a ₁	Análisis	3
e ₂	María	112	Matanzas	a ₂	Lógica	3
e ₄	Rita	112	Mayabeque	a ₂	Lógica	3
e ₄	Rita	112	Mayabeque	a ₄	Programación	4
e ₅	Carlos	113	Pinar del Río	a ₃	Álgebra	3

Todas las tuplas deben ser modificadas en una misma transacción

¿Es este un buen diseño? (Redundancia)

<u>#E</u>	ENombre	Grupo	Provincia	<u>#A</u>	ANombre	Nota
e ₁	Juan	111	La Habana	a ₁	Análisis	3
e ₁	Juan	111	La Habana	a ₂	Lógica	2
e ₁	Juan	111	La Habana	a ₃	Álgebra	4
e ₁	Juan	111	La Habana	a ₄	Programación	5
e ₃	Pedro	111	La Habana	a ₃	Álgebra	4
e ₂	María	112	Matanzas	a ₁	Análisis	3
e ₂	María	112	Matanzas	a ₂	Lógica	3
e ₄	Rita	112	Mayabeque	a ₂	Lógica	3
e ₄	Rita	112	Mayabeque	a ₄	Programación	4
e ₅	Carlos	113	Pinar del Río	a ₃	Álgebra	3

¿Es necesaria esta redundancia?

Entonces...

¿Cómo solucionar estas anomalías?

Estudiante		
<u>#E</u>	ENombre	Provincia
e_1	Juan	La Habana
e_2	María	Matanzas
e_3	Pedro	La Habana
e_4	Rita	Mayabeque
e_5	Carlos	Pinar del Río

Provincia-Grupo	
<u>Provincia</u>	Grupo
La Habana	111
Matanzas	112
Mayabeque	112
Pinar del Río	113

Asignatura	
<u>#A</u>	ANombre
a_1	Análisis
a_2	Lógica
a_3	Álgebra
a_4	Programación

Evaluar		
<u>#E</u>	<u>#A</u>	Nota
e_1	a_1	3
e_1	a_2	2
e_1	a_3	4
e_1	a_4	5
e_3	a_3	4
e_2	a_1	3
e_2	a_2	3
e_4	a_2	3
e_4	a_4	4
e_5	a_3	3

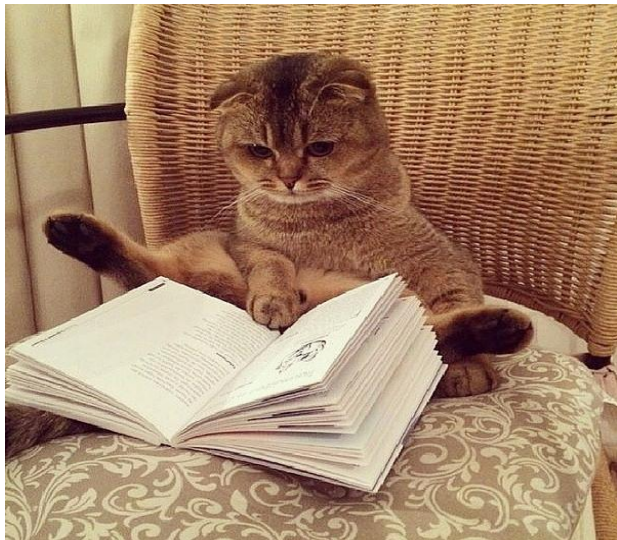
¿Cómo obtener esta solución?

¿Cómo obtener esta solución?

continuará...

Entonces...

... alguna duda?



Bases de Datos

Dependencias Funcionales y Anomalías

Lic. Andy Ledesma García
Lic. Víctor M. Cardentey Fundora
Dra. C. Lucina García Hernández

Departamento de Computación
Facultad de Matemática y Computación
Universidad de La Habana

Licenciatura en Ciencia de Datos

13 de febrero de 2024

Anexos

Anexos

Algoritmo para determinar si un conjunto de atributos cumple la unicidad

Entrada: $U = \{A_1, A_2, \dots, A_n\}$, F conjunto de dependencias funcionales y X , $X \subseteq U$

Salida: 1 si el conjunto X cumple la unicidad en $R(U, F)$ o 0 en otro caso.

Método:

1. Sea X el conjunto de atributos que se desea comprobar. Primero inicializamos $X_0 = X$.
2. En cada iteración i se busca una dependencia funcional $Y \rightarrow A$ tal que $Y \subseteq X_{i-1}$, pero $A \notin X_{i-1}$. Entonces se asigna $X_i = X_{i-1} \cup \{A\}$.
3. Repetir el paso 2 tantas veces como sea necesario hasta que no puedan añadirse más atributos. Dado que el conjunto resultante solo puede crecer y la cantidad de atributos en el universo es finito, eventualmente el algoritmo termina.
4. Sea k la iteración final del algoritmo, se comprueba que $X_k = U$.