Bases de Datos I

Teoría del diseño: dependencias funcionales

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

28 de mayo de 2024

¿Qué es una llave candidata?

¿Qué es una llave candidata?

Llave candidata

Un conjunto de uno o más atributos $K = \{A_1, A_2, ..., A_n\}$ 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 $A_1, A_2, ..., A_n$.
2. **Minimalidad**: Ningún subconjunto propio de K tiene la propiedad de unicidad.

¿Cómo encontrar las llaves candidatas de una relación?

Objetivos

1. Establecer un marco matemático-computacional que permita definir y aplicar un conjunto de restricciones de integridad sobre una base de datos relacional.
2. Utilizar el marco teórico definido para desarrollar algoritmos que permitan determinar las llaves candidatas y la equivalencia entre relaciones.

¿Cómo encontrar las llaves candidatas de una relación?

La idea es muy fácil

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

Minimalidad

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

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

Minimalidad

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

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

Para comprobar la minimalidad debemos comprobar la unicidad

Unicidad

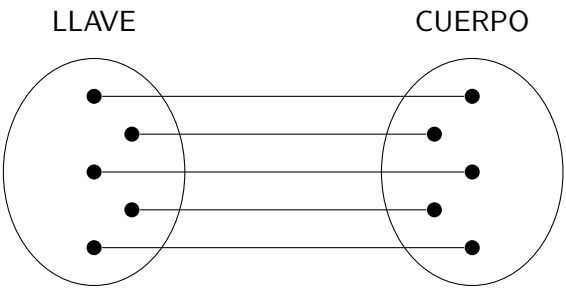
¿Qué es la unicidad?

Unicidad

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

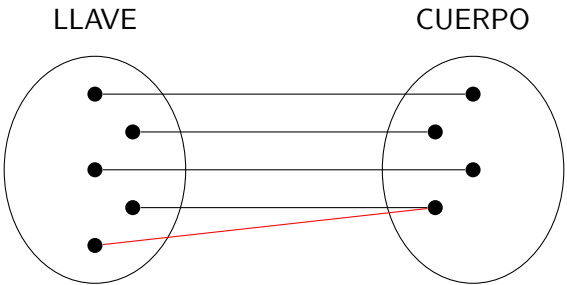
Unicidad

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



Unicidad

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



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.

$$\text{Email} \rightarrow \text{Nombre, Apellido}$$

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.

$$\text{Email} \rightarrow \text{Nombre, Apellido}$$

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

$$\text{C. Postal} \rightarrow \text{Provincia, País}$$

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.

$$\text{Email} \rightarrow \text{Nombre, Apellido}$$

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

$$\text{C. Postal} \rightarrow \text{Provincia, País}$$

¿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 solo 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

Sean $U = A_1, ..., A_n$ el universo de atributos de la relación R, $X \subseteq U$ y $Y \subseteq U$. La dependencia funcional $X \rightarrow Y$ se cumple en R si para toda instancia $r(R)$ para todos los pares de tuplas t_1 y t_2 en r se cumple que:

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

¿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.
- 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

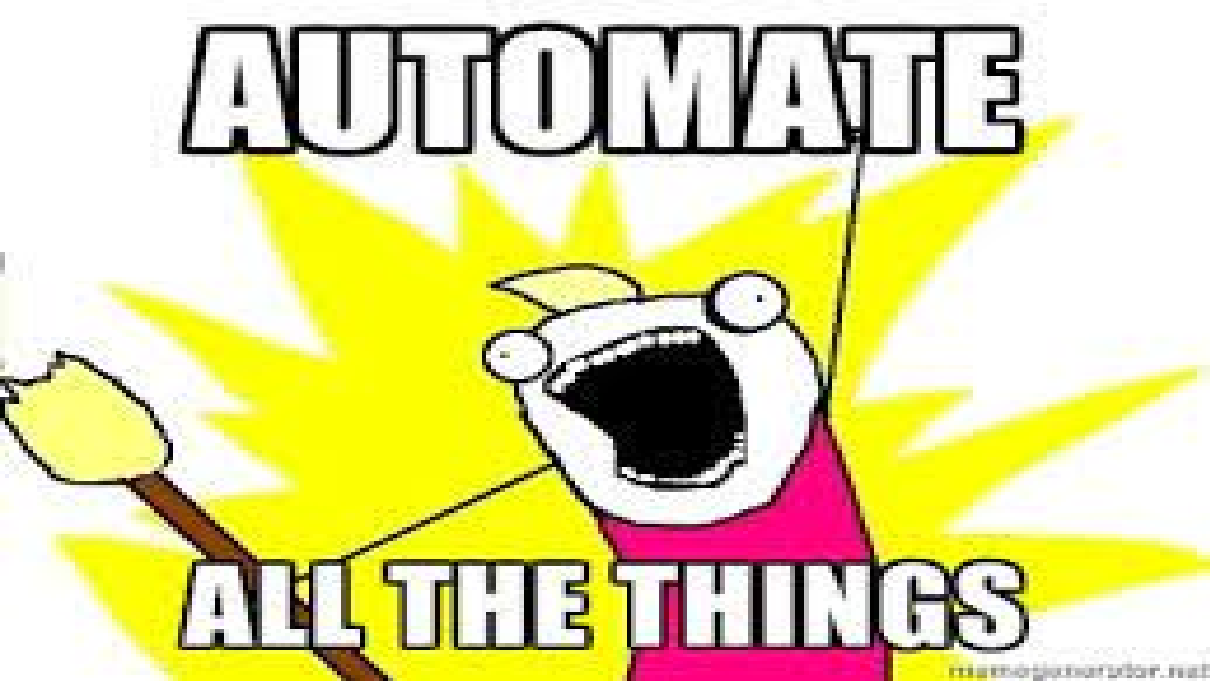
- Especificar las restricciones en el conjunto de instancias legales de una relación R (instancias r de R que satisfacen un conjunto de dependencias funcionales F):

  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

Algoritmo para determinar si un conjunto de atributos cumple la unicidad

Entrada: $U = \{A_1, A_2, ..., A_n\}$, 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:

Algoritmo para determinar si un conjunto de atributos cumple la unicidad

Entrada: $U = \{A_1, A_2, ..., A_n\}$, 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 $X_0 = X$.

Algoritmo para determinar si un conjunto de atributos cumple la unicidad

Entrada: $U = \{A_1, A_2, ..., A_n\}$, 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 $X_i = X_{i-1} \cup \{A\}$.

Algoritmo para determinar si un conjunto de atributos cumple la unicidad

Entrada: $U = \{A_1, A_2, ..., A_n\}$, 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.

Algoritmo para determinar si un conjunto de atributos cumple la unicidad

Entrada: $U = \{A_1, A_2, ..., A_n\}$, 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 $X_k = U$.

Ejemplo

Sea $R(U, F)$ con:

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

comprobar si CD cumple la unicidad.

Ejemplo

Sea $R(U, F)$ con:

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

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

Ejemplo

Sea $R(U, F)$ con:

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

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

Ejemplo

Sea $R(U, F)$ con:

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

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

1. $X_1 = ACD$

Ejemplo

Sea $R(U, F)$ con:

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

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

1. $X_1 = ACD$

Ejemplo

Sea $R(U, F)$ con:

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

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

1. $X_1 = ACD$
2. $X_2 = ABCD$

Ejemplo

Sea $R(U, F)$ con:

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

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

1. $X_1 = ACD$
2. $X_2 = ABCD$

Ejemplo

Sea $R(U, F)$ con:

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

comprobar si CD cumple la unicidad.

Inicializamos $X_0 = CD$

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

Ejemplo

Sea $R(U, F)$ con:

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

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

Formalizando

Implicación lógica

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

$$F \models X \to Y$$

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 $\{A_1, A_2, ..., A_n\}$ de una esquema relacional $R(U, F)$ es llave candidata 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 solo 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.

Propiedades de los axiomas

Sea un esquema relacional $R(U, F)$ y $X \to Y$ una dependencia funcional:

- **Sólidos**: Si $X \to Y$ se deduce de F aplicando los axiomas de Armstrong, entonces $X \to Y$ es verdadera en cualquier instancia de $R(U, F)$.
- **Completos**: Si $X \to Y$ es verdadera en cualquier instancia de $R(U, F)$, entonces $X \to Y$ se deduce lógicamente de F aplicando los Axiomas de Armstrong.

Lemas derivados

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

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

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:

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

Clasificando dependencias funcionales

Sea una dependencia funcional $A_1, A_2, ..., A_n \rightarrow B_1, B_2, ..., B_m$:

- Es trivial si $\{B_1, B_2, ..., B_m\} \subseteq \{A_1, A_2, ..., A_n\}$
- Es no trivial si existe $B_i \notin \{A_1, A_2, ..., A_n\}$.
- Es completamente no trivial si $\{B_1, B_2, ..., B_m\} \cap \{A_1, A_2, ..., A_n\} = \emptyset$

Clasificando dependencias funcionales

Sea una dependencia funcional $A_1, A_2, ..., A_n \rightarrow B_1, B_2, ..., B_m$:

- Es trivial si $\{B_1, B_2, ..., B_m\} \subseteq \{A_1, A_2, ..., A_n\}$
- Es no trivial si existe $B_i \notin \{A_1, A_2, ..., A_n\}$.
- Es completamente no trivial si $\{B_1, B_2, ..., B_m\} \cap \{A_1, A_2, ..., A_n\} = \emptyset$

Ejemplos

- Trivial: $A \rightarrow A$
- No trivial: $A \rightarrow AB$
- Completamente no trivial: $A \rightarrow B$

¿Es viable computar F^+?

¿Cuántas dependencias funcionales triviales existen en F^+?

¿Es viable computar F^+?

¿Cuántas dependencias funcionales triviales existen en F^+?

- Sea un conjunto de atributos X se tiene que $X \rightarrow X'$, $\forall X' : \ X' \subseteq X, X' \neq \emptyset$

¿Es viable computar F^+?

¿Cuántas dependencias funcionales triviales existen en F^+?

- Sea un conjunto de atributos X se tiene que $X \rightarrow X'$, $\forall X' : \ X' \subseteq X, X' \neq \emptyset$
- El conjunto de DF's triviales cuyo determinante es X tiene $2^{|X|} - 1$ elementos.

¿Es viable computar F^+?

¿Cuántas dependencias funcionales triviales existen en F^+?

- Sea un conjunto de atributos X se tiene que $X \rightarrow X'$, $\forall X' : \ X' \subseteq X, X' \neq \emptyset$
- El conjunto de DF's triviales cuyo determinante es X tiene $2^{|X|} - 1$ elementos.
- 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^+?

- Sea un conjunto de atributos X se tiene que $X \rightarrow X'$, $\forall X' : \ X' \subseteq X, X' \neq \emptyset$
- El conjunto de DF's triviales cuyo determinante es X tiene $2^{|X|} - 1$ elementos.
- 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 $Y \subseteq X_F^+$

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 correctitud de la base de datos?

Entonces...

... alguna duda?