

Bases de Datos

Diseño Lógico: Modelo relacional. Diseño Intuitivo

Lic. Andy Ledesma García

Lic. Víctor M. Cardentey Fundora

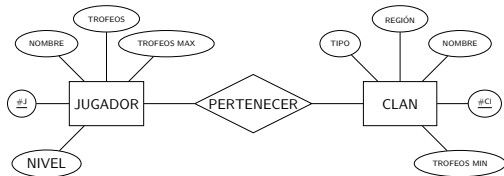
Dra. C. Lucina García Hernández

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

Licenciatura en Ciencia de Datos

6 de febrero de 2024

¿Dónde nos quedamos?



JUGADOR

#J	Nombre	Nivel	Trofeos	TrofeosMax
1	Juan	13	7500	7560
2	Pedro	11	7000	7200
3	María	12	7050	7400
.
.

CLAN

#C	Nombre	Región	Tipo	TrofeosMin
1	River Plate 2.	MEX	Cerrado	7000
2	TheWarriors	GER	Invitación	7300
3	WestRoyale	ESP	Cerrado	6300
.
.

PERTENECER

#J	#C
1	2
2	3
3	1
.	.
.	.

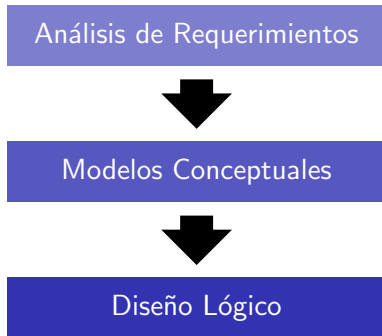
Fases del diseño de una base de datos

Análisis de Requerimientos



Modelos Conceptuales

Fases del diseño de una base de datos



Diseño lógico

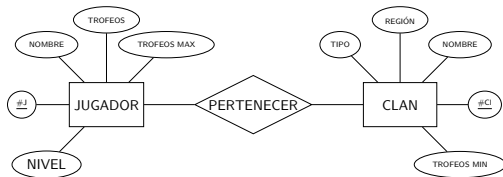
El diseño lógico es una descripción de las estructuras de datos utilizadas para almacenar los datos.

Entonces...

Lo que usted aprenderá

1. Identificar un modelo matemático de datos y sus componentes principales.
2. Aplicar el modelo relacional
3. Obtener un diseño intuitivo de una base de datos relacional

Representación tabular



JUGADOR

<u>#J</u>	Nombre	Nivel	Trofeos	TrofeosMax
1	Juan	13	7500	7560
2	Pedro	11	7000	7200
3	María	12	7050	7400
.
.

CLAN

<u>#C</u>	Nombre	Región	Tipo	TrofeosMin
1	River Plate 2.	MEX	Cerrado	7000
2	TheWarriors	GER	Invitación	7300
3	WestRoyale	ESP	Cerrado	6300
.
.

PERTENECER

<u>#J</u>	<u>#C</u>
1	2
2	3
3	1
.	.
.	.

¿Cómo podemos describir un MERX?

¿Cómo podemos describir un MERX?

Modelo matemático de datos

Un modelo matemático de datos es una definición lógica, abstracta y auto-contenida de:

¿Cómo podemos describir un MERX?

Modelo matemático de datos

Un modelo matemático de datos es una definición lógica, abstracta y auto-contenida de:

- ▶ **Estructuras de datos:** utilizadas para la representación de los datos y sus interrelaciones.

¿Cómo podemos describir un MERX?

Modelo matemático de datos

Un modelo matemático de datos es una definición lógica, abstracta y auto-contenida de:

- ▶ **Estructuras de datos:** utilizadas para la representación de los datos y sus interrelaciones.
- ▶ **Restricciones de integridad:** utilizadas para mantener un estado consistente de la base de datos durante la ejecución de operaciones que modifican la base de datos.

¿Cómo podemos describir un MERX?

Modelo matemático de datos

Un modelo matemático de datos es una definición lógica, abstracta y auto-contenida de:

- ▶ **Estructuras de datos:** utilizadas para la representación de los datos y sus interrelaciones.
- ▶ **Restricciones de integridad:** utilizadas para mantener un estado consistente de la base de datos durante la ejecución de operaciones que modifican la base de datos.
- ▶ **Operaciones:** utilizadas para manipular los datos.

¿Cómo podemos describir un MERX?

Modelo matemático de datos

Un modelo matemático de datos es una definición lógica, abstracta y auto-contenida de:

- ▶ **Estructuras de datos:** utilizadas para la representación de los datos y sus interrelaciones.
- ▶ **Restricciones de integridad:** utilizadas para mantener un estado consistente de la base de datos durante la ejecución de operaciones que modifican la base de datos.
- ▶ **Operaciones:** utilizadas para manipular los datos.

que integradas constituyen una máquina abstracta con la que los usuarios interactúan.

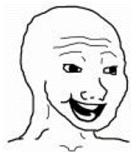
La implementación no es una descripción

Implementación de un modelo matemático de datos

Es una realización física en una máquina real de los componentes de la máquina abstracta que constituye el modelo.

Todo es acerca de la conveniencia

Usuario de SGBD



Lenguaje declarativo

Desarrollador de SGBD



Estructuras de datos

Algoritmos

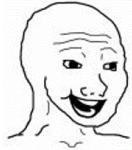
Optimización

Compilación

Gestión de ficheros

Todo es acerca de la conveniencia

Usuario de SGBD



Modelo matemático de datos

Desarrollador de SGBD



Implementación

USUARIO

**ENFOQUE
PRE-RELACIONAL**

IMPLEMENTACIÓN

**MODELO
MATEMÁTICO DE DATOS**

QUÉ \$@#!% ES ESTO

Enfoques pre-relacionales

Importancia

- ▶ Fueron las primeras soluciones computacionales capaces de almacenar y consultar grandes conjuntos de datos.
- ▶ Se desarrollaron productos comerciales de larga vida basados en estos sistemas.

Enfoques pre-relacionales

Importancia

- ▶ Fueron las primeras soluciones computacionales capaces de almacenar y consultar grandes conjuntos de datos.
- ▶ Se desarrollaron productos comerciales de larga vida basados en estos sistemas.

Problemas

- ▶ Los modelos de datos se consideran como abstracciones de las estructuras de almacenamiento subyacentes en el nivel físico y sus operadores asociados.

Enfoques pre-relacionales

Importancia

- ▶ Fueron las primeras soluciones computacionales capaces de almacenar y consultar grandes conjuntos de datos.
- ▶ Se desarrollaron productos comerciales de larga vida basados en estos sistemas.

Problemas

- ▶ El modelo dependía de la implementación.

Enfoques pre-relacionales

Importancia

- ▶ Fueron las primeras soluciones computacionales capaces de almacenar y consultar grandes conjuntos de datos.
- ▶ Se desarrollaron productos comerciales de larga vida basados en estos sistemas.

Problemas

- ▶ El modelo dependía de la implementación.
- ▶ Los datos se representan por colecciones de registros (*records*) y las interrelaciones entre los datos se representan mediante enlaces (*links*).

Enfoques pre-relacionales

Importancia

- ▶ Fueron las primeras soluciones computacionales capaces de almacenar y consultar grandes conjuntos de datos.
- ▶ Se desarrollaron productos comerciales de larga vida basados en estos sistemas.

Problemas

- ▶ El modelo dependía de la implementación.
- ▶ Eran muy complicados de utilizar.

Enfoques pre-relacionales

Importancia

- ▶ Fueron las primeras soluciones computacionales capaces de almacenar y consultar grandes conjuntos de datos.
- ▶ Se desarrollaron productos comerciales de larga vida basados en estos sistemas.

Problemas

- ▶ El modelo dependía de la implementación.
- ▶ Eran muy complicados de utilizar.
- ▶ Los usuarios son programadores que se deben encargar, incluso, de la optimización.

Modelos jerárquico y reticular

Modelo jerárquico

- ▶ Las interrelaciones se representan como jerarquías.

Modelo jerárquico

- ▶ Las interrelaciones se representan como jerarquías.
- ▶ Ningún hijo puede existir sin su padre.

Modelos jerárquico y reticular

Modelo jerárquico

- ▶ Las interrelaciones se representan como jerarquías.
- ▶ Ningún hijo puede existir sin su padre.
- ▶ Se recorre un árbol para: insertar, actualizar, eliminar y buscar.

Modelos jerárquico y reticular

Modelo jerárquico

- ▶ Las interrelaciones se representan como jerarquías.
- ▶ Ningún hijo puede existir sin su padre.
- ▶ Se recorre un árbol para: insertar, actualizar, eliminar y buscar.

Modelo reticular

- ▶ Las interrelaciones se representan a través de un grafo orientado.

Modelos jerárquico y reticular

Modelo jerárquico

- ▶ Las interrelaciones se representan como jerarquías.
- ▶ Ningún hijo puede existir sin su padre.
- ▶ Se recorre un árbol para: insertar, actualizar, eliminar y buscar.

Modelo reticular

- ▶ Las interrelaciones se representan a través de un grafo orientado.
- ▶ No tiene restricciones de integridad.

Modelos jerárquico y reticular

Modelo jerárquico

- ▶ Las interrelaciones se representan como jerarquías.
- ▶ Ningún hijo puede existir sin su padre.
- ▶ Se recorre un árbol para: insertar, actualizar, eliminar y buscar.

Modelo reticular

- ▶ Las interrelaciones se representan a través de un grafo orientado.
- ▶ No tiene restricciones de integridad.
- ▶ Se recorre un grafo para: insertar, actualizar, eliminar y buscar.

Modelos jerárquico y reticular

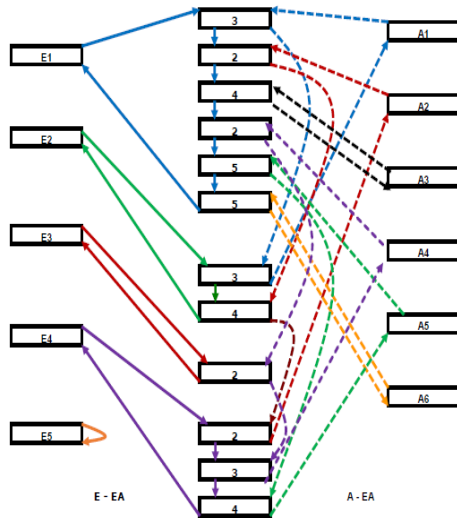
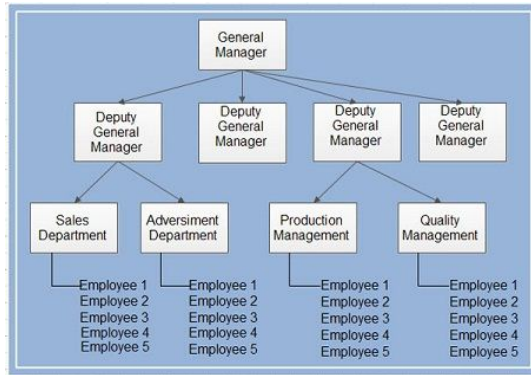
Modelo jerárquico

- ▶ Las interrelaciones se representan como jerarquías.
- ▶ Ningún hijo puede existir sin su padre.
- ▶ Se recorre un árbol para: **insertar**, **actualizar**, **eliminar** y **buscar**.

Modelo reticular

- ▶ Las interrelaciones se representan a través de un grafo orientado.
- ▶ No tiene restricciones de integridad.
- ▶ Se recorre un grafo para: **insertar**, **actualizar**, **eliminar** y **buscar**.

Una imagen vale más que mis palabras



Propuesta de E. F. Codd (1970)

- ▶ Relacionar los datos mediante vínculos naturales, lógicos, inherentes a los datos y al fenómeno y no a su representación computacional.
- ▶ Lograr un modelo simple en el que tanto los datos como los vínculos que se establecen entre ellos se representen mediante tablas.

¿Están listos chicos?



Estructura de datos

¿Cómo describir una tabla?

¿Qué conceptos matemáticos pudiesen modelar una tabla?

JUGADOR

<u>#J</u>	Nombre	Nivel	Trofeos	TrofeosMax
1	Juan	13	7500	7560
2	Pedro	11	7000	7200
3	María	12	7050	7400
⋮	⋮	⋮	⋮	⋮

¿Nos servirá alguna estructura matemática que ya conocemos?

¿Podemos utilizar una matriz?

$$A_{m \times n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

¿Nos servirá alguna estructura matemática que ya conocemos?

¿Podemos utilizar una matriz?

$$A_{m \times n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

No. Las matrices se definen sobre un único dominio

¿Nos servirá alguna estructura matemática que ya conocemos?

Dominio

Conjunto de valores que puede tomar un atributo.

¿Nos servirá alguna estructura matemática que ya conocemos?

Dominio

Conjunto de valores que puede tomar un atributo.

Relación (Teoría de conjuntos)

La relación n -aria sobre los dominios D_1, D_2, \dots, D_n es el conjunto de tuplas ordenadas (a_1, a_2, \dots, a_n) pertenecientes al producto cartesiano $D_1 \times D_2 \times \dots \times D_n$, donde $a_i \in D_i$, para cada $i \in 1, \dots, n$, cuya condición $R(a_1, a_2, \dots, a_n)$ se satisface.

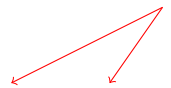
$$R = \{(a_1, a_2, \dots, a_n) \in D_1 \times D_2 \times \dots \times D_n \mid R(a_1, a_2, \dots, a_n)\}$$

¿Se podría mejorar?

$$\text{JUGADOR} = \left\{ \begin{array}{l} < 1, \text{Juan}, 13, 7500, 7560 >, \\ < 2, \text{Pedro}, 11, 7000, 7200 >, \\ < 3, \text{María}, 12, 7050, 7400 >, \\ \vdots \end{array} \right\}$$

¿Se podría mejorar?

¿Cómo el usuario distingue entre el identificador y el nivel?



$$\text{JUGADOR} = \left\{ \begin{array}{l} < 1, \text{Juan}, 13, 7500, 7560 >, \\ < 2, \text{Pedro}, 11, 7000, 7200 >, \\ < 3, \text{María}, 12, 7050, 7400 >, \\ \vdots \end{array} \right\}$$

Problemas

- ▶ No es autodescriptiva

¿Se podría mejorar?

El usuario debe recordar que el nombre es el segundo elemento


$$\text{JUGADOR} = \left\{ \begin{array}{l} < 1, \text{Juan}, 13, 7500, 7560 >, \\ < 2, \text{Pedro}, 11, 7000, 7200 >, \\ < 3, \text{María}, 12, 7050, 7400 >, \\ \vdots \end{array} \right\}$$

Problemas

- ▶ No es autodescriptiva
- ▶ El orden de los datos importa

¿Cómo usar relaciones para almacenar registros?

Relación (Bases de datos)

Una relación R definida sobre un conjunto de dominios D_1, D_2, \dots, D_n , no necesariamente distintos, se compone de:

- La **cabecera**, formada por un conjunto finito de pares atributo-dominio

$$\{(A_1 : D_1), (A_2 : D_2), \dots, (A_n : D_n)\}$$

tal que el atributo A_j corresponde al y sólo al dominio D_j para todo $j = 1, 2, \dots, n$.

¿Cómo usar relaciones para almacenar registros?

Relación (Bases de datos)

Una relación R definida sobre un conjunto de dominios D_1, D_2, \dots, D_n , no necesariamente distintos, se compone de:

- La **cabecera**, formada por un conjunto finito de pares atributo-dominio

$$\{(A_1 : D_1), (A_2 : D_2), \dots, (A_n : D_n)\}$$

tal que el atributo A_j corresponde al y sólo al dominio D_j para todo $j = 1, 2, \dots, n$.

Cabecera para la relación Jugador

$$\{(\#J : \mathbb{N}), (\text{Nombre} : \mathbb{S}), (\text{Nivel} : \mathbb{N}), (\text{Trofeos} : \mathbb{N}), (\text{TrofeosMax} : \mathbb{N})\}$$

$$\mathbb{S} = \{\text{Conjunto de todas las cadenas de longitud } \leq 20\}$$

¿Cómo usar relaciones para almacenar registros?

Relación (Bases de datos)

Una relación R definida sobre un conjunto de dominios D_1, D_2, \dots, D_n , no necesariamente distintos, se compone de:

- El **cuerpo**, está formado por un conjunto finito de tuplas, el cual varía en el tiempo. Cada tupla, a su vez, está formada por un conjunto de pares atributo-valor.

$$\{(A_1 : V_{i1}), (A_2 : V_{i2}), \dots, (A_n : V_{in})\}, \quad (i = 1, 2, \dots, m)$$

tal que m es la cantidad de tuplas en el conjunto y $V_{ij} \in D_j$ para todo par $(A_j : V_{ij})$ con $j = 1, 2, \dots, n$

¿Cómo usar relaciones para almacenar registros?

Relación (Bases de datos)

Una relación R definida sobre un conjunto de dominios D_1, D_2, \dots, D_n , no necesariamente distintos, se compone de:

- El **cuerpo**, está formado por un conjunto finito de tuplas, el cual varía en el tiempo. Cada tupla, a su vez, está formada por un conjunto de pares atributo-valor.

$$\{(A_1 : V_{i1}), (A_2 : V_{i2}), \dots, (A_n : V_{in})\}, (i = 1, 2, \dots, m)$$

tal que m es la cantidad de tuplas en el conjunto y $V_{ij} \in D_j$ para todo par $(A_j : V_{ij})$ con $j = 1, 2, \dots, n$

Ejemplo de tupla para la relación Jugador

$$\{(\#J : 1), (\text{Nombre} : \text{Juan}), (\text{Nivel} : 13), (\text{Trofeos} : 7500), (\text{TrofeosMax} : 7560)\}$$

Identificando registros

Llave candidata

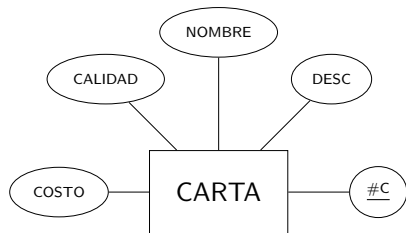
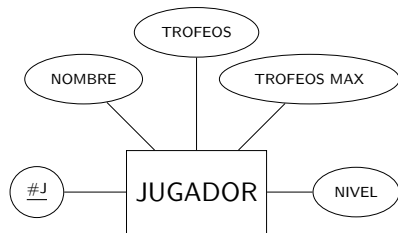
Un conjunto de uno o más atributos $K = \{A_1, A_2, \dots, A_n\}$ es una llave candidata de la relación R si cumple las siguientes propiedades:

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.

Llave primaria

Es una de las llaves candidatas que se selecciona como llave de la relación.

Representando relaciones



Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

Carta(#C, Nombre, Calidad, Desc, Costo)

Relaciones para almacenar interrelaciones

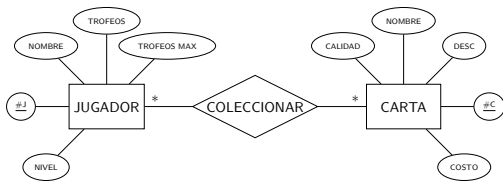
Llave foránea

Un conjunto de uno o más atributos $F = \{A_1, A_2, \dots, A_n\}$ de una relación R , correspondientes a los dominios D_1, D_2, \dots, D_n respectivamente, es una llave foránea referente a la relación R' si:

1. La llave primaria de R' es un conjunto de atributos $P = \{B_1, B_2, \dots, B_n\}$ correspondientes a los dominios D_1, D_2, \dots, D_n respectivamente.
2. Existe un acuerdo de correspondencia entre los atributos A_i y B_i para todo $i = 1, 2, \dots, n$

Una tupla $t \in R$ referencia a una tupla $t' \in R'$ si el valor de A_i en la tupla t es igual al valor de B_i en la tupla t' para todo $i = 1, 2, \dots, n$.

Interrelacionando registros: ejemplo



Carta(#C, Nombre, Calidad, Desc, Costo)

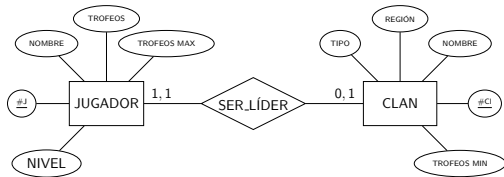
Coleccionar(#J, #C)

FK: #J REFERENCIA A Jugador

FK: #C REFERENCIA A Carta

Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

Interrelacionando registros: ejemplo



Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

Clan(#Cl, #Líder, Nombre, Tipo, Región, TrofeosMin)

FK: #Líder REFERENCIA A Jugador (#J)

Identificando relaciones

Esquema de una relación

El esquema de una relación es una especificación de su estructura, la cual es independiente de las tuplas que contiene el cuerpo. El esquema se compone de:

- ▶ El nombre de la relación
- ▶ La cabecera
- ▶ La llave primaria
- ▶ Las llaves foráneas

En una misma base de datos una relación se identifica unívocamente por su nombre.

Identificando relaciones

Esquema de una relación

El esquema de una relación es una especificación de su estructura, la cual es independiente de las tuplas que contiene el cuerpo. El esquema se compone de:

- ▶ El nombre de la relación
- ▶ La cabecera
- ▶ La llave primaria
- ▶ Las llaves foráneas

En una misma base de datos una relación se identifica unívocamente por su nombre.

Instancia de una relación

Se refiere al conjunto de tuplas que constituye el cuerpo de la relación en un momento específico del tiempo.

Identificando relaciones: ejemplo

Relación R

$\{(\underline{A}:\mathbb{N}), (B:\mathbb{N}), (C:\mathbb{N})\}$		
$\{(A:2), (B:5), (C:6)\}$		
$\{(A:4), (B:7), (C:8)\}$		
$\{(A:9), (B:10), (C:11)\}$		

Relación R

$\{(\underline{A}:\mathbb{N}), (B:\mathbb{N}), (C:\mathbb{N})\}$		
$\{(A:2), (B:5), (C:6)\}$		
$\{(A:4), (B:7), (C:8)\}$		
$\{(A:9), (B:10), (C:11)\}$		
$\{(A:1), (B:12), (C:3)\}$		

→
Añadir tupla $\{(A:1), (B:12), (C:3)\}$

Misma relación. Diferentes instancias

Modelo relacional

Estructura de datos

Relación

Restricciones de integridad

¿Qué es un estado consistente de la base de datos?

Estado de una base de datos

- Conjunto de instancias $\{r_1, r_2, \dots, r_n\}$ de las relaciones R_1, R_2, \dots, R_n respectivamente que conforman la base de datos en un instante de tiempo específico.

¿Qué es un estado consistente de la base de datos?

Estado de una base de datos

- ▶ Conjunto de instancias $\{r_1, r_2, \dots, r_n\}$ de las relaciones R_1, R_2, \dots, R_n respectivamente que conforman la base de datos en un instante de tiempo específico.
- ▶ Un estado es consistente si satisface cada una de las restricciones de integridad definidas sobre la base de datos.

Integridad de las entidades

Todos los atributos de una llave primaria deben ser no nulos

Relación R

{(<u>A</u> : \mathbb{N}), (B: \mathbb{N}), (C: \mathbb{N})}		
{(A:1), (B:2022), (C:1000)}		
{(A:1), (B:2021), (C:1000)}		
{(A:2), (B:2022), (C:1200)}		

Integridad de las entidades

Todos los atributos de una llave primaria deben ser no nulos

Relación R

{(<u>A</u> :N), (B:N), (C:N)}		
{(A:1), (B:2022), (C:1000)}		
{(A:1), (B:2021), (C:1000)}		
{(A:2), (B:2022), (C:1200)}		

¿Se puede insertar esta tupla?
{(A:1),(B:NULL),(C:1000)}

Integridad de las entidades

Todos los atributos de una llave primaria deben ser no nulos

Relación R

{(<u>A</u> :N), (<u>B</u> :N), (C:N)}		
{(A:1), (B:2022), (C:1000)}		
{(A:1), (B:2021), (C:1000)}		
{(A:2), (B:2022), (C:1200)}		

¿Se puede insertar esta tupla?
{(A:1), (B:NULL), (C:1000)}

La minimalidad

Integridad de las entidades

Todos los atributos de una llave primaria deben ser no nulos

Relación R

{(<u>A</u> : \mathbb{N}), (<u>B</u> : \mathbb{N}), (C: \mathbb{N})}		
{(A:1), (B:2022), (C:1000)}		
{(A:1), (B:2021), (C:1000)}		
{(A:2), (B:2022), (C:1200)}		

¿Se puede insertar esta tupla?
{(A:1), (B:NULL), (C:1000)}

La llave es minimal. Si su valor está incompleto entonces no es único

Integridad referencial

- ▶ Todos los atributos de una llave foránea deben ser no nulos o todos deben ser nulos.
- ▶ El valor de una llave foránea tiene que ser un valor existente de la llave primaria en la relación a la que hace referencia.

Integridad referencial

- ▶ Todos los atributos de una llave foránea deben ser no nulos o todos deben ser nulos.
- ▶ El valor de una llave foránea tiene que ser un valor existente de la llave primaria en la relación a la que hace referencia.

Relación R

$\{(\underline{A}:\mathbb{N}), (\underline{B}:\mathbb{N}), (C:\mathbb{N})\}$		
$\{(A:1), (B:2022), (C:1000)\}$		
$\{(A:1), (B:2021), (C:1000)\}$		
$\{(A:2), (B:2022), (C:1200)\}$		

Relación S

FK: (A,B) REFERENCES R (A,B)

$\{(\underline{D}:\mathbb{N}), (A:\mathbb{N}), (B:\mathbb{N})\}$		
$\{(D:2), (A:2), (B:2022)\}$		

La minimalidad... otra vez

Integridad referencial

- ▶ Todos los atributos de una llave foránea deben ser no nulos o todos deben ser nulos.
- ▶ El valor de una llave foránea tiene que ser un valor existente de la llave primaria en la relación a la que hace referencia.

Relación R

{(<u>A</u> : \mathbb{N}), (<u>B</u> : \mathbb{N}), (C: \mathbb{N})}		
{(A:1), (B:2022), (C:1000)}		
{(A:1), (B:2021), (C:1000)}		
{(A:2), (B:2022), (C:1200)}		

Relación S

FK: (A,B) REFERENCES R (A,B)

{(<u>D</u> : \mathbb{N}), (A: \mathbb{N}), (B: \mathbb{N})}		
{(D:2), (A: NULL), (B:2022)}		

Integridad referencial

- ▶ Todos los atributos de una llave foránea deben ser no nulos o todos deben ser nulos.
- ▶ El valor de una llave foránea tiene que ser un valor existente de la llave primaria en la relación a la que hace referencia.

Relación R

{(<u>A</u> : \mathbb{N}), (<u>B</u> : \mathbb{N}), (C: \mathbb{N})}		
{(A:1), (B:2022), (C:1000)}		
{(A:1), (B:2021), (C:1000)}		
{(A:2), (B:2022), (C:1200)}		

Relación S

FK: (A,B) REFERENCES R (A,B)

{(<u>D</u> : \mathbb{N}), (A: \mathbb{N}), (B: \mathbb{N})}		
{(D:2), (A:NULL), (B:2022)}		

¿A cuál tupla de la relación R referencia la tupla de la relación S?

Integridad referencial

- ▶ Todos los atributos de una llave foránea deben ser no nulos o todos deben ser nulos.
- ▶ El valor de una llave foránea tiene que ser un valor existente de la llave primaria en la relación a la que hace referencia.

Relación R

{(<u>A</u> : \mathbb{N}), (<u>B</u> : \mathbb{N}), (C: \mathbb{N})}		
{(A:1), (B:2022), (C:1000)}		
{(A:1), (B:2021), (C:1000)}		
{(A:2), (B:2022), (C:1200)}		

Relación S

FK: (A,B) REFERENCES R (A,B)

{(<u>D</u> : \mathbb{N}), (A: \mathbb{N}), (B: \mathbb{N})}		
{(D:2), (A:NULL), (B:2022)}		

El valor de una llave foránea es un valor de la llave primaria de otra tabla
También le afecta la minimalidad.

Integridad de los dominios

- ▶ Todos los valores de un atributo de una relación tienen que provenir del dominio pertinente.

Integridad de los dominios

- ▶ Todos los valores de un atributo de una relación tienen que provenir del dominio pertinente.




Trivial

Modelo relacional

Estructura de datos

Relación

Restricciones de integridad

- ▶ Metarreglas 
- ▶ Dependencias funcionales (*spoiler*) 
- ▶ Restricciones del negocio (*spoiler*) 

Operaciones

WHAT IF I TOLD YOU THAT



**RELATIONAL DATABASES
ARE IN FACT EVENT STORES**

Un modelo transaccional

Transacciones

Una transacción es un conjunto de operaciones que modifican el estado de la base de datos y:

Un modelo transaccional

Transacciones

Una transacción es un **conjunto de operaciones que modifican el estado de la base de datos** y:

- ▶ **Atomicity**: Se considera como una sola operación, es decir, se realizan todos los cambios o no se realiza ninguno.

Un modelo transaccional

Transacciones

Una transacción es un **conjunto de operaciones que modifican el estado de la base de datos** y:

- ▶ **A***tomicity*: Se considera como una sola operación, es decir, se realizan todos los cambios o no se realiza ninguno.
- ▶ **C***onsistency*: El estado de la base de datos es consistente antes y después de ejecutarse la transacción, pudiendo no serlo durante la ejecución.

Un modelo transaccional

Transacciones

Una transacción es un **conjunto de operaciones que modifican el estado de la base de datos** y:

- ▶ **Atomicity**: Se considera como una sola operación, es decir, se realizan todos los cambios o no se realiza ninguno.
- ▶ **Consistency**: El estado de la base de datos es consistente antes y después de ejecutarse la transacción, pudiendo no serlo durante la ejecución.
- ▶ **Isolation**: El estado intermedio de una transacción no es visible por el resto de las transacciones: se aísla.

Un modelo transaccional

Transacciones

Una transacción es un **conjunto de operaciones que modifican el estado de la base de datos** y:

- ▶ **Atomicity**: Se considera como una sola operación, es decir, se realizan todos los cambios o no se realiza ninguno.
- ▶ **Consistency**: El estado de la base de datos es consistente antes y después de ejecutarse la transacción, pudiendo no serlo durante la ejecución.
- ▶ **Isolation**: El estado intermedio de una transacción no es visible por el resto de las transacciones: se aísla.
- ▶ **Durability**: Luego de ejecutada la transacción, los cambios de estado son persistentes y no pueden ser deshechos, incluso, en el caso de fallos del sistema.

Un modelo transaccional

Transacciones

Una transacción es un **conjunto de operaciones que modifican el estado de la base de datos** y:

- ▶ **Atomicity**: Se considera como una sola operación, es decir, se realizan todos los cambios o no se realiza ninguno.
- ▶ **Consistency**: El estado de la base de datos es consistente antes y después de ejecutarse la transacción, pudiendo no serlo durante la ejecución.
- ▶ **Isolation**: El estado intermedio de una transacción no es visible por el resto de las transacciones: se aísla.
- ▶ **Durability**: Luego de ejecutada la transacción, los cambios de estado son persistentes y no pueden ser deshechos, incluso, en el caso de fallos del sistema.

Propiedades ACID

Operaciones que modifican el estado

Operaciones que modifican el estado

Insertar

1. Debemos comprobar que todas las metarreglas se cumplan para la tupla que vamos a insertar.

Operaciones que modifican el estado

Insertar

1. Debemos comprobar que todas las metarreglas se cumplan para la tupla que vamos a insertar.

Eliminar

Operaciones que modifican el estado

Insertar

1. Debemos comprobar que todas las metarreglas se cumplan para la tupla que vamos a insertar.

Eliminar

1. No tenemos que comprobar las metarreglas en la tupla que vamos a eliminar pues ya está insertada

Operaciones que modifican el estado

Insertar

1. Debemos comprobar que todas las metarreglas se cumplan para la tupla que vamos a insertar.

Eliminar

1. No tenemos que comprobar las metarreglas en la tupla que vamos a eliminar pues ya está insertada

¿Y si eliminamos un valor de llave primaria que es el valor de una llave foránea?

Operaciones que modifican el estado

Insertar

1. Debemos comprobar que todas las metarreglas se cumplan para la tupla que vamos a insertar.

Eliminar

1. No tenemos que comprobar las metarreglas en la tupla que vamos a eliminar pues ya está insertada
2. La relación debe avisar al resto de las relaciones que un valor de la llave primaria se va a eliminar

Operaciones que modifican el estado

Insertar

1. Debemos comprobar que todas las metarreglas se cumplan para la tupla que vamos a insertar.

Eliminar

1. No tenemos que comprobar las metarreglas en la tupla que vamos a eliminar pues ya está insertada
2. La relación debe avisar al resto de las relaciones que un valor de la llave primaria se va a eliminar
3. Las relaciones que referencian a la tupla eliminada tienen tres opciones: detener la eliminación, anular la llave foránea o eliminar las tuplas que utilicen dicho valor de llave foránea.



**OMG! I ran DELETE operation on
production DB**

8548 rows affected

Operaciones que modifican el estado

Actualizar

Operaciones que modifican el estado

Actualizar

Se puede ver como un proceso de insertar una tupla y eliminar una tupla.

Operaciones que modifican el estado

Actualizar

1. Debemos hacer las comprobaciones pertinentes a la inserción de la tupla modificada.

Operaciones que modifican el estado

Actualizar

1. Debemos hacer las comprobaciones pertinentes a la inserción de la tupla modificada.
2. Si el valor de la llave primaria se modifica debemos avisar al resto de las relaciones para que actualicen la llave foránea.

Operaciones que modifican el estado

Actualizar

1. Debemos hacer las comprobaciones pertinentes a la inserción de la tupla modificada.
2. Si el valor de la llave primaria se modifica debemos avisar al resto de las relaciones para que actualicen la llave foránea.
3. Realizar una transacción de dos operaciones:
 - 3.1 Eliminar la tupla original
 - 3.2 Insertar la tupla modificada

Operaciones para responder preguntas

Álgebra relacional

PROFE...



ÁLGEBRA OTRA VEZ?

Operaciones para responder preguntas

Álgebra relacional

- ▶ Lenguaje procedimental para el manejo y construcción de relaciones.
- ▶ Compuesta por operandos (variables y relaciones) y operadores (extensión de la Teoría de conjuntos).
- ▶ Las operaciones del álgebra relacional manipulan y producen relaciones (Propiedad relacional del cierre o clausura).

Asignación ($:=$)

La asignación se utiliza para destinar a una variable de relación el valor que se crea a partir de la aplicación de cualesquiera de las operaciones sobre las relaciones existentes.

Renombrar

Operación que no afecta el conjunto de tuplas presentes en la relación sino que modifica el esquema de la relación. En específico permite cambiar el nombre tanto de los atributos como el de la relación.

Operaciones de Teoría de conjuntos: Ejemplo

Relación R

$\{(A:\mathbb{N}) \quad (B:\mathbb{N})\}$
$\{(A:1), (B:2)\}$
$\{(A:2), (B:3)\}$
$\{(A:3), (B:4)\}$
$\{(A:4), (B:4)\}$

Relación S

$\{(A:\mathbb{N}) \quad (B:\mathbb{N})\}$
$\{(A:1), (B:2)\}$
$\{(A:2), (B:4)\}$
$\{(A:5), (B:7)\}$
$\{(A:6), (B:1)\}$

Relación $R \cup S$

$\{(A:\mathbb{N}) \quad (B:\mathbb{N})\}$
$\{(A:1), (B:2)\}$
$\{(A:2), (B:3)\}$
$\{(A:3), (B:4)\}$
$\{(A:4), (B:4)\}$
$\{(A:5), (B:7)\}$
$\{(A:6), (B:1)\}$

Relación $R \cap S$

$\{(A:\mathbb{N}) \quad (B:\mathbb{N})\}$
$\{(A:1), (B:2)\}$
$\{(A:2), (B:3)\}$

Relación $R - S$

$\{(A:\mathbb{N}) \quad (B:\mathbb{N})\}$
$\{(A:3), (B:4)\}$
$\{(A:4), (B:4)\}$

Restricción: Ejemplo

Relación R

$\{(\underline{A}:\mathbb{N})$	$(B:\mathbb{N})$	$(C:\mathbb{N})\}$
$\{(A:1),$	$(B:2),$	$(C:5)\}$
$\{(A:2),$	$(B:2),$	$(C:7)\}$
$\{(A:3),$	$(B:4),$	$(C:7)\}$
$\{(A:4),$	$(B:4),$	$(C:7)\}$

Relación $R \sigma (B = 4 \wedge C = 7)$

$\{(\underline{A}:\mathbb{N})$	$(B:\mathbb{N})$	$(C:\mathbb{N})\}$
$\{(A:3),$	$(B:4),$	$(C:7)\}$
$\{(A:4),$	$(B:4),$	$(C:7)\}$

Relación $R \sigma (B + C > 7)$

$\{(\underline{A}:\mathbb{N})$	$(B:\mathbb{N})$	$(C:\mathbb{N})\}$
$\{(A:2),$	$(B:2),$	$(C:7)\}$
$\{(A:3),$	$(B:4),$	$(C:7)\}$
$\{(A:4),$	$(B:4),$	$(C:7)\}$

Relación $R \sigma (B = 2)$

$\{(\underline{A}:\mathbb{N})$	$(B:\mathbb{N})$	$(C:\mathbb{N})\}$
$\{(A:1),$	$(B:2),$	$(C:5)\}$
$\{(A:2),$	$(B:2),$	$(C:7)\}$

Proyección: Ejemplo

Relación R

$\{(A:\mathbb{N})$	$(B:\mathbb{N})$	$(C:\mathbb{N})\}$
$\{(A:1),$	$(B:2),$	$(C:5)\}$
$\{(A:2),$	$(B:2),$	$(C:7)\}$
$\{(A:3),$	$(B:4),$	$(C:7)\}$
$\{(A:4),$	$(B:4),$	$(C:7)\}$

Relación $\pi_{A,B}(R)$

$\{(A:\mathbb{N})$	$(B:\mathbb{N})\}$
$\{(A:1),$	$(B:2)\}$
$\{(A:2),$	$(B:2)\}$
$\{(A:3),$	$(B:4)\}$
$\{(A:4),$	$(B:4)\}$

Relación $\pi_B(R)$

$\{(B:\mathbb{N})\}$
$\{(B:2)\}$
$\{(B:4)\}$

Relación $\pi_{B,C}(R)$

$\{(B:\mathbb{N})$	$(C:\mathbb{N})\}$
$\{(B:2),$	$(C:5)\}$
$\{(B:2),$	$(C:7)\}$
$\{(B:4),$	$(C:7)\}$

Producto Cartesiano: Ejemplo

Relación R

$\{(\underline{A}:\mathbb{N}), (B:\mathbb{N})\}$
$\{(A:1), (B:2)\}$
$\{(A:3), (B:4)\}$

Relación S

$\{(\underline{B}:\mathbb{N}), (C:\mathbb{N}), (D:\mathbb{N})\}$
$\{(B:2), (C:5), (D:6)\}$
$\{(B:4), (C:7), (D:8)\}$
$\{(B:9), (C:10), (D:11)\}$

Producto Cartesiano: Ejemplo

Relación R

$\{(\underline{A}:\mathbb{N}), (B:\mathbb{N})\}$
$\{(A:1), (B:2)\}$
$\{(A:3), (B:4)\}$

Relación S

$\{(B:\mathbb{N}), (C:\mathbb{N}), (D:\mathbb{N})\}$
$\{(B:2), (C:5), (D:6)\}$
$\{(B:4), (C:7), (D:8)\}$
$\{(B:9), (C:10), (D:11)\}$

Relación $R \times S$

$\{(\underline{A}:\mathbb{N})$	$(R.B:\mathbb{N})$	$(\underline{S.B}:\mathbb{N}),$	$(C:\mathbb{N}),$	$(D:\mathbb{N})\}$
$\{(A:1),$	$(R.B:2),$	$(S.B:2),$	$(C:5),$	$(D:6)\}$
$\{(A:1),$	$(R.B:2),$	$(S.B:4),$	$(C:7),$	$(D:8)\}$
$\{(A:1),$	$(R.B:2),$	$(S.B:9),$	$(C:10),$	$(D:11)\}$
$\{(A:3),$	$(R.B:4),$	$(S.B:2),$	$(C:5),$	$(D:6)\}$
$\{(A:3),$	$(R.B:4),$	$(S.B:4),$	$(C:7),$	$(D:8)\}$
$\{(A:3),$	$(R.B:4),$	$(S.B:9),$	$(C:10),$	$(D:11)\}$

Ya tenemos todas las operaciones que necesitamos...

Ya tenemos todas las operaciones que necesitamos...

...para poder definir el resto

Operaciones que combinan relaciones

Natural Join ($R \bowtie S$)

Sean R y S dos relaciones, no necesariamente distintas, distinguimos los siguientes conjuntos de atributos:

- ▶ El conjunto R_1, \dots, R_m son atributos de R y no de S .
- ▶ El conjunto S_1, \dots, S_n son atributos de S y no de R .
- ▶ El conjunto A_1, \dots, A_k son atributos comunes de R y S , es decir, atributos con el mismo nombre y mismo dominio asociado en ambas relaciones.

La operación $R \bowtie S$ se define como:

$$\pi_{R_1, \dots, R_m, S_1, \dots, S_n, R.A_1, \dots, R.A_k}((R \times S) \sigma(R.A_1 = S.A_1 \wedge \dots \wedge R.A_k = S.A_k))$$

Operaciones que combinan relaciones

Natural Join ($R \bowtie S$)

Sean R y S dos relaciones, no necesariamente distintas, distinguimos los siguientes conjuntos de atributos:

- ▶ El conjunto R_1, \dots, R_m son atributos de R y no de S .
- ▶ El conjunto S_1, \dots, S_n son atributos de S y no de R .
- ▶ El conjunto A_1, \dots, A_k son atributos comunes de R y S , es decir, atributos con el mismo nombre y mismo dominio asociado en ambas relaciones.

La operación $R \bowtie S$ ($R \text{ JOIN } S$) se define como:

$$\pi_{R_1, \dots, R_m, S_1, \dots, S_n, R.A_1, \dots, R.A_k}((R \times S) \sigma(R.A_1 = S.A_1 \wedge \dots \wedge R.A_k = S.A_k))$$

Natural Join: Ejemplo

Relación R

$\{(\underline{A}:\mathbb{N}), (B:\mathbb{N})\}$
$\{(A:1), (B:2)\}$
$\{(A:3), (B:4)\}$

Relación S

$\{(B:\mathbb{N}), (C:\mathbb{N}), (D:\mathbb{N})\}$
$\{(B:2), (C:5), (D:6)\}$
$\{(B:4), (C:7), (D:8)\}$
$\{(B:9), (C:10), (D:11)\}$

Relación $R \times S$

$\{(\underline{A}:\mathbb{N}) (R.B:\mathbb{N}) (\underline{S.B}:\mathbb{N}), (C:\mathbb{N}), (D:\mathbb{N})\}$
$\{(A:1), (R.B:2), (S.B:2), (C:5), (D:6)\}$
$\{(A:1), (R.B:2), (S.B:4), (C:7), (D:8)\}$
$\{(A:1), (R.B:2), (S.B:9), (C:10), (D:11)\}$
$\{(A:3), (R.B:4), (S.B:2), (C:5), (D:6)\}$
$\{(A:3), (R.B:4), (S.B:4), (C:7), (D:8)\}$
$\{(A:3), (R.B:4), (S.B:9), (C:10), (D:11)\}$

Natural Join: Ejemplo

Relación $R \times S$

$\{(\underline{A}:\mathbb{N})$	$(R.B:\mathbb{N})$	$(\underline{S.B}:\mathbb{N}),$	$(C:\mathbb{N}),$	$(D:\mathbb{N})\}$
$\{(A:1),$	$(R.B:2),$	$(S.B:2),$	$(C:5),$	$(D:6)\}$
$\{(A:1),$	$(R.B:2),$	$(S.B:4),$	$(C:7),$	$(D:8)\}$
$\{(A:1),$	$(R.B:2),$	$(S.B:9),$	$(C:10),$	$(D:11)\}$
$\{(A:3),$	$(R.B:4),$	$(S.B:2),$	$(C:5),$	$(D:6)\}$
$\{(A:3),$	$(R.B:4),$	$(S.B:4),$	$(C:7),$	$(D:8)\}$
$\{(A:3),$	$(R.B:4),$	$(S.B:9),$	$(C:10),$	$(D:11)\}$

Natural Join: Ejemplo

$$F : R.B = S.B$$

Relación $R \times S$

$\{(\underline{A}:\mathbb{N})$	$(R.B:\mathbb{N})$	$(\underline{S.B}:\mathbb{N}),$	$(C:\mathbb{N}),$	$(D:\mathbb{N})\}$
$\{(A:1),$	$(R.B:2),$	$(S.B:2),$	$(C:5),$	$(D:6)\}$
$\{(A:1),$	$(R.B:2),$	$(S.B:4),$	$(C:7),$	$(D:8)\}$
$\{(A:1),$	$(R.B:2),$	$(S.B:9),$	$(C:10),$	$(D:11)\}$
$\{(A:3),$	$(R.B:4),$	$(S.B:2),$	$(C:5),$	$(D:6)\}$
$\{(A:3),$	$(R.B:4),$	$(S.B:4),$	$(C:7),$	$(D:8)\}$
$\{(A:3),$	$(R.B:4),$	$(S.B:9),$	$(C:10),$	$(D:11)\}$

Natural Join: Ejemplo

$$F : R.B = S.B$$

Relación $R \times S$

$\{(A:\mathbb{N})$	$(R.B:\mathbb{N})$	$(S.B:\mathbb{N}),$	$(C:\mathbb{N}),$	$(D:\mathbb{N})\}$
$\{(A:1),$	$(R.B:2),$	$(S.B:2),$	$(C:5),$	$(D:6)\}$
$\{(A:1),$	$(R.B:2),$	$(S.B:4),$	$(C:7),$	$(D:8)\}$
$\{(A:1),$	$(R.B:2),$	$(S.B:9),$	$(C:10),$	$(D:11)\}$
$\{(A:3),$	$(R.B:4),$	$(S.B:2),$	$(C:5),$	$(D:6)\}$
$\{(A:3),$	$(R.B:4),$	$(S.B:4),$	$(C:7),$	$(D:8)\}$
$\{(A:3),$	$(R.B:4),$	$(S.B:9),$	$(C:10),$	$(D:11)\}$

Natural Join: Ejemplo

$$F : R.B = S.B$$

Relación $(R \times S) \sigma F$

$\{(\underline{A}:\mathbb{N})$	$(R.B:\mathbb{N})$	$(\underline{S.B}:\mathbb{N}),$	$(C:\mathbb{N}),$	$(D:\mathbb{N})\}$
$\{(A:1),$	$(R.B:2),$	$(S.B:2),$	$(C:5),$	$(D:6)\}$
$\{(A:3),$	$(R.B:4),$	$(S.B:4),$	$(C:7),$	$(D:8)\}$

Natural Join: Ejemplo

Relación $\pi_{A,R.B,C,D} ((R \times S) \sigma F)$

$\{(\underline{A}:\mathbb{N}) \quad (\underline{R.B}:\mathbb{N}) \quad (C:\mathbb{N}), \quad (D:\mathbb{N})\}$
$\{(A:1), \quad (R.B:2), \quad (C:5), \quad (D:6)\}$
$\{(A:3), \quad (R.B:4), \quad (C:7), \quad (D:8)\}$

Natural Join: Ejemplo

Relación $R \bowtie S$

{(<u>A</u> : \mathbb{N}) (<u>B</u> : \mathbb{N}) (C: \mathbb{N}), (D: \mathbb{N})}			
{(A:1),	(B:2),	(C:5),	(D:6)}
{(A:3),	(B:4),	(C:7),	(D:8)}

Operaciones que combinan relaciones

Theta Join (θ -Join)

Sean R y S dos relaciones, no necesariamente distintas, definimos el θ -Join de R y S como:

$$(R \times S) \sigma F : F = \theta$$

donde θ es un condición, expresada mediante una fórmula bien formada.

Modelo relacional

Estructura de datos

Relación

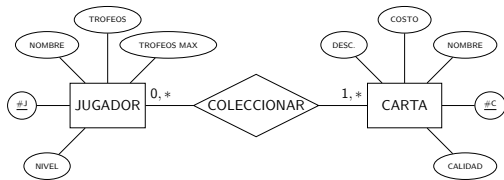
Restricciones de integridad

- ▶ Metarreglas
- ▶ Dependencias funcionales (Spoiler)
- ▶ Restricciones del negocio (Spoiler)

Operaciones

- ▶ Insertar
- ▶ Actualizar
- ▶ Eliminar
- ▶ Álgebra relacional

¿Qué queremos?



Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

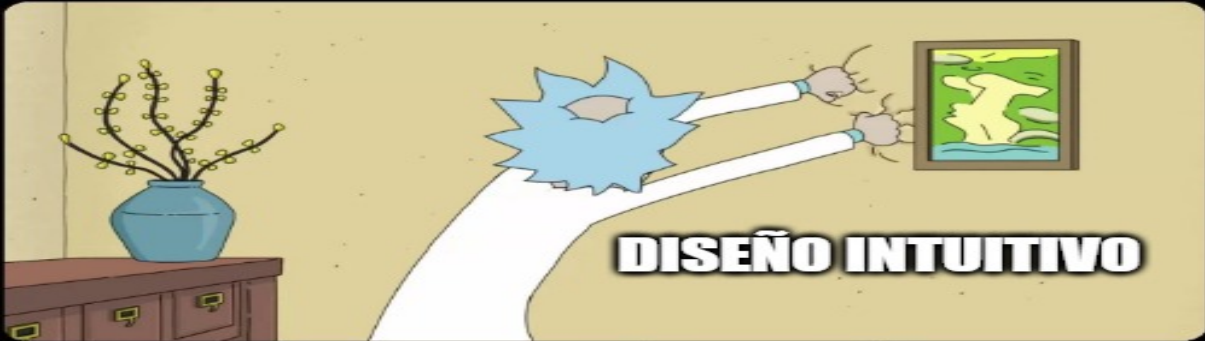
Carta(#C, Nombre, Calidad, Desc., Costo)

Coleccionar(#J, #C)

FK: #J REFERENCES Jugador

FK: #C REFERENCES Carta

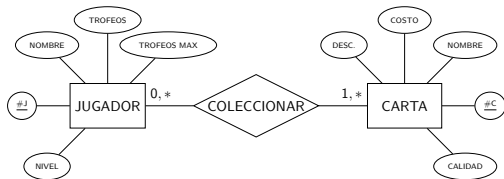
Diseño intuitivo



Ahora sí... transformando el diseño

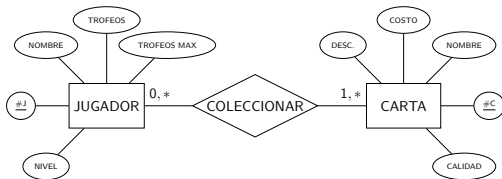
Algoritmo del diseño intuitivo

Diseño intuitivo



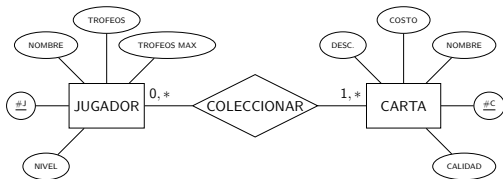
La idea básica

1. Convertir cada conjunto de entidades en una relación con el mismo conjunto de atributos.



La idea básica

1. Convertir cada conjunto de entidades en una relación con el mismo conjunto de atributos.

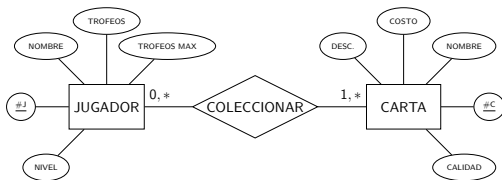


Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

Carta(#C, Nombre, Calidad, Desc., Costo)

La idea básica

1. Convertir cada conjunto de entidades en una relación con el mismo conjunto de atributos.
2. Convertir cada interrelación en una relación cuyos atributos son las llaves primarias de las relaciones que representan los conjuntos de entidades conectados.

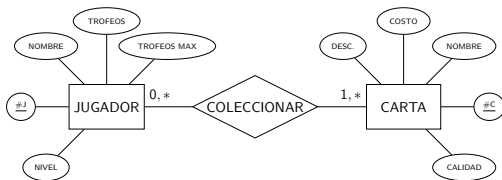


Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

Carta(#C, Nombre, Calidad, Desc., Costo)

La idea básica

1. Convertir cada conjunto de entidades en una relación con el mismo conjunto de atributos.
2. Convertir cada interrelación en una relación cuyos atributos son las llaves primarias de las relaciones que representan los conjuntos de entidades conectados.



Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

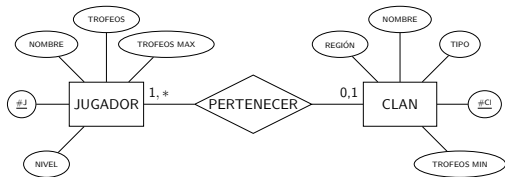
Carta(#C, Nombre, Calidad, Desc., Costo)

Coleccionar(#J, #C)

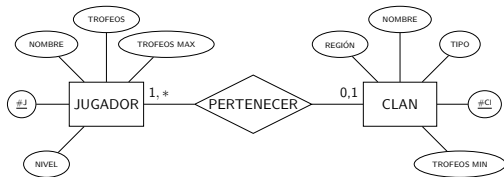
FK: #J REFERENCES Jugador

FK: #C REFERENCES Carta

Aplicando el algoritmo



Aplicando el algoritmo



Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

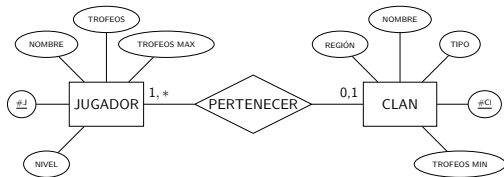
Clan(#CI, Nombre, Región, Tipo, TrofeosMin)

Pertenecer(#J, #CI)

FK: #J REFERENCES Jugador

FK: #CI REFERENCES Clan

Aplicando el algoritmo



Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

Clan(#CI, Nombre, Región, Tipo, TrofeosMin)

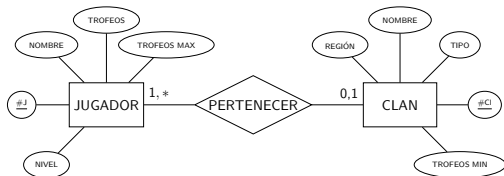
Pertenecer(#J, #CI)

FK: #J REFERENCES Jugador

FK: #CI REFERENCES Clan

¿Este diseño es eficiente?

Un caso especial



Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax, #CI)

FK: #CI REFERENCES Clan HAS NULL

Clan(#CI, Nombre, Región, Tipo, TrofeosMin)

Añadir la llave primaria de la relación correspondiente al conjunto de entidades en el extremo con cardinalidad máxima **uno** a la relación correspondiente al conjunto de entidades en el otro extremo.

Diseños para interrelaciones de muchos a uno

Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

Clan(#CI, Nombre, Región, Tipo, TrofeosMin)

Pertenecer(#J, #CI)

FK: #J REFERENCES Jugador

FK: #CI REFERENCES Clan

Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax, #CI)

FK: #CI REFERENCES Clan HAS NULL

Clan(#CI, Nombre, Región, Tipo, TrofeosMin)

Diseños para interrelaciones de muchos a uno

Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

Clan(#CI, Nombre, Región, Tipo, TrofeosMin)

Pertenecer(#J, #CI)

FK: #J REFERENCES Jugador

FK: #CI REFERENCES Clan

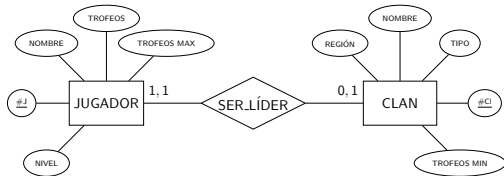
Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax, #CI)

FK: #CI REFERENCES Clan HAS NULL

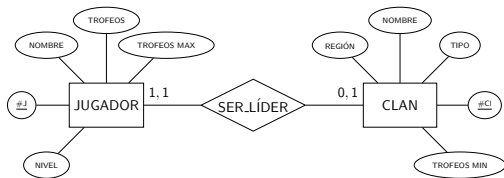
Clan(#CI, Nombre, Región, Tipo, TrofeosMin)

TAREA: ¿Cuándo uno es mejor que el otro?

El caso especial del caso especial



Diseño para interrelaciones uno a uno: Opcionalidad

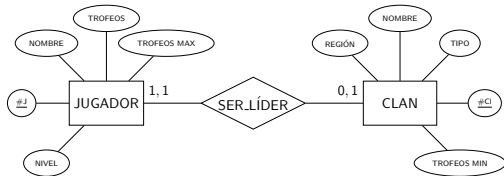


Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax, #CI)

FK: #CI REFERENCES Clan HAS NULL

Clan(#CI, Nombre, Región, Tipo, TrofeosMin)

Diseño para interrelaciones uno a uno: Obligatoriedad



Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

Clan(#Cl, Nombre, Región, Tipo, TrofeosMin, #J)

FK: #J REFERENCES Jugador

Diseño para interrelaciones de uno a uno

Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax, #Cl)

FK: #Cl REFERENCES Clan HAS NULL

Clan(#Cl, Nombre, Región, Tipo, TrofeosMin)

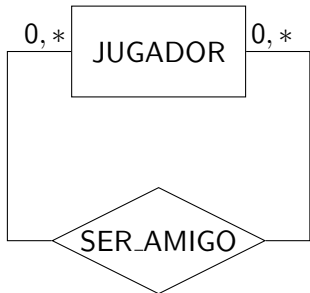
Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

Clan(#Cl, Nombre, Región, Tipo, TrofeosMin, #J)

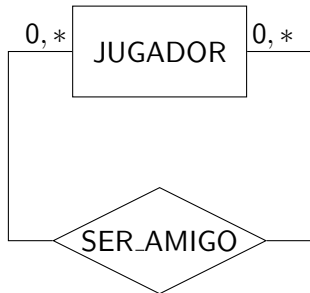
FK: #J REFERENCES Jugador

Seleccionar el extremo con mayor cardinalidad mínima es más eficiente

Diseño para interrelaciones con roles



Diseño para interrelaciones con roles

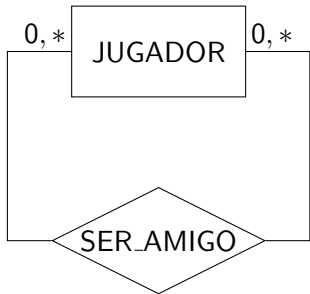


Ser_Amigo(#Amigo1, #Amigo2)

FK: #Amigo1 REFERENCES Jugador (#J)

FK: #Amigo2 REFERENCES Jugador (#J)

Diseño para interrelaciones con roles



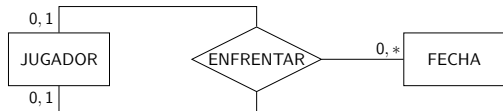
Ser_Amigo(#Amigo1, #Amigo2)

FK: #Amigo1 REFERENCES Jugador (#J)

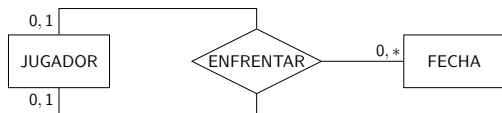
FK: #Amigo2 REFERENCES Jugador (#J)

Se deben renombrar los atributos que tienen el mismo nombre

Diseño para interrelaciones n -arias



Diseño para interrelaciones n -arias



Fecha(Fecha)

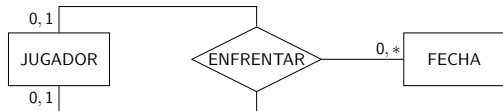
Enfrentar(#J1, #J2, Fecha)

FK: #J1 REFERENCES Jugador (#J)

FK: #J2 REFERENCES Jugador (#J)

FK: Fecha REFERENCES Fecha

Diseño para interrelaciones n -arias



Fecha(Fecha)

Enfrentar(#J1, #J2, Fecha)

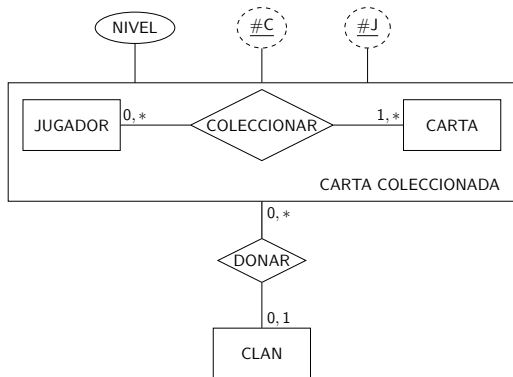
FK: #J1 REFERENCES Jugador (#J)

FK: #J2 REFERENCES Jugador (#J)

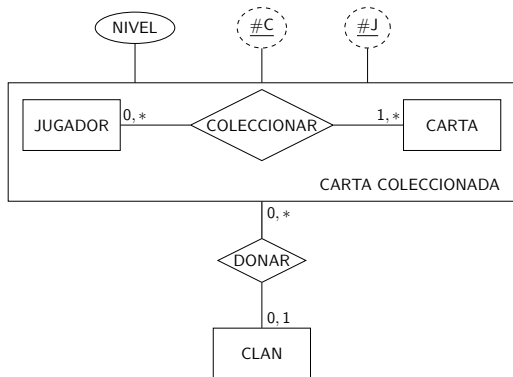
FK: Fecha REFERENCES Fecha

- ▶ Convertir la interrelación en una relación cuyos atributos son las llaves primarias de las relaciones que representan los conjuntos de entidades conectados.
- ▶ Si existen extremos de la interrelación con cardinalidad máxima uno, se escoge uno de ellos y su llave primaria se retira de la llave primaria de la relación resultante.

Diseño de agregaciones



Diseño de agregaciones



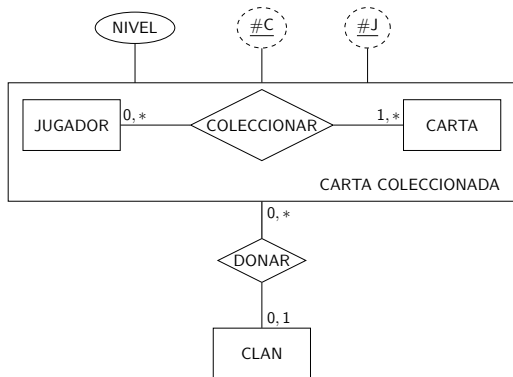
Coleccionar(#J, #C, Nivel)

Donar(#J, #C, #CI)

FK: (#J, #C) REFERENCES Coleccionar (#J, #C)

FK: #CI REFERENCES Clan

Diseño de agregaciones



Coleccionar(#J, #C, Nivel)

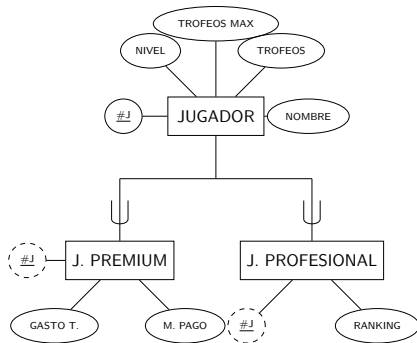
Donar(#J, #C, #CI)

FK: (#J, #C) REFERENCES Coleccionar (#J, #C)

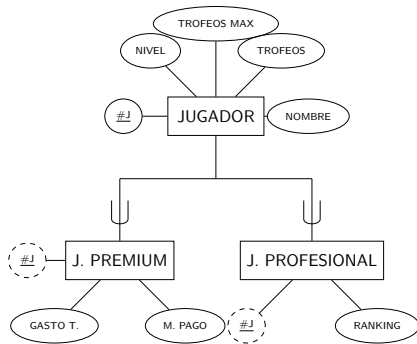
FK: #CI REFERENCES Clan

Agregar los atributos a la relación resultante de la interrelación

Diseños para la especialización

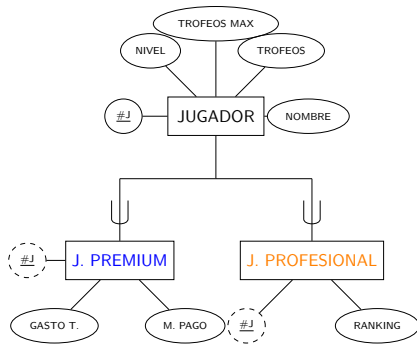


Diseños para la especialización



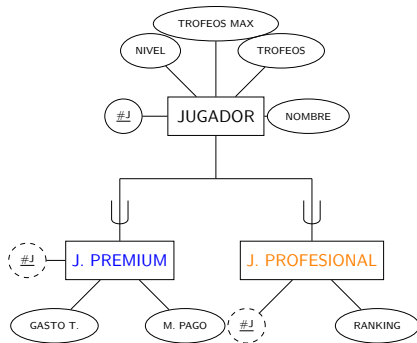
Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax, Gasto T., M. Pago, Ranking)

Diseños para la especialización



Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax,
Gasto T., M. Pago, Ranking)

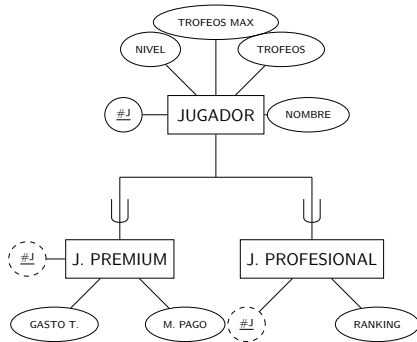
Diseños para la especialización



Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax,
Gasto T., M. Pago, Ranking)

¿Qué ocurre si un jugador es profesional pero no premium?

Diseños para la especialización



Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

J. Premium(#J, Gasto T., M. Pago)

FK: #J REFERENCES Jugador

J. Profesional(#J, Ranking)

FK: #J REFERENCES Jugador

Diseños para la especialización

Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax,
Gasto T., M. Pago, Ranking)

Jugador(#J, Nombre, Nivel, Trofeos, TrofeosMax)

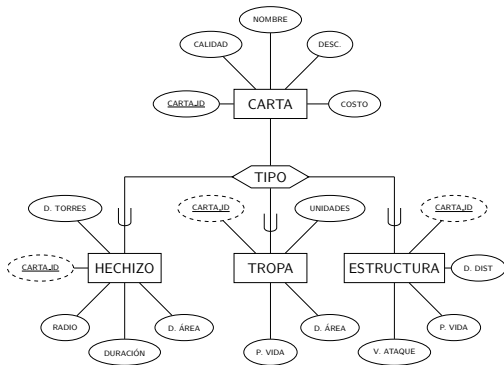
J. Premium(#J, Gasto T., M. Pago)

FK: #J REFERENCES Jugador

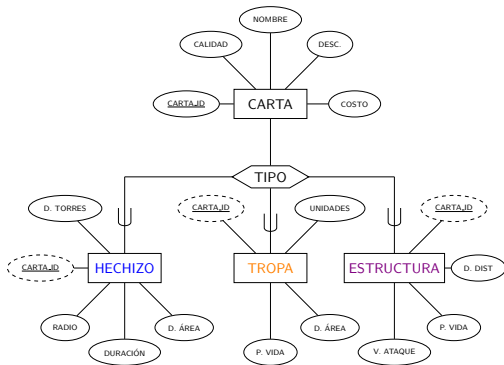
J. Profesional(#J, Ranking)

FK: #J REFERENCES Jugador

Diseños para la especialización (partición)

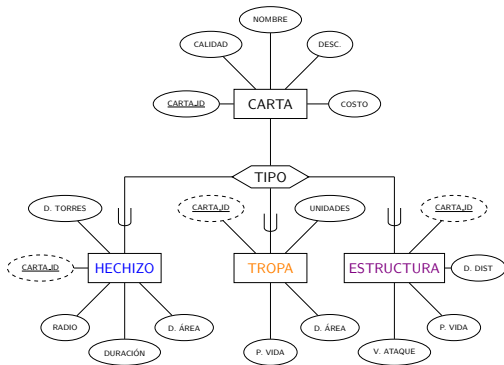


Diseños para la especialización (partición)



Carta(#C, Nombre, Calidad, Desc., Costo, D.Torres, Radio, Duración, H. D. Área, Unidades, T. P. Vida, T. D. Área, D. Dist, E. P. Vida, V. Ataque)

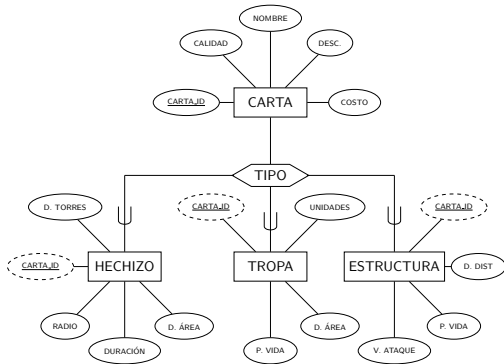
Diseños para la especialización (partición)



Carta(#C, Nombre, Calidad, Desc., Costo, **D. Torres**,
Radio, **Duración**, **H. D. Área**, **Unidades**, **T. P. Vida**,
T. D. Área, **D. Dist**, **E. P. Vida**, **V. Ataque**)

Muy ineficiente espacialmente

Diseños para la especialización (partición)



Carta(#C, Nombre, Calidad, Desc., Costo)

Hechizo(#C, D.Torres, Radio, Duración, D. Área)

FK: #C REFERENCES Carta

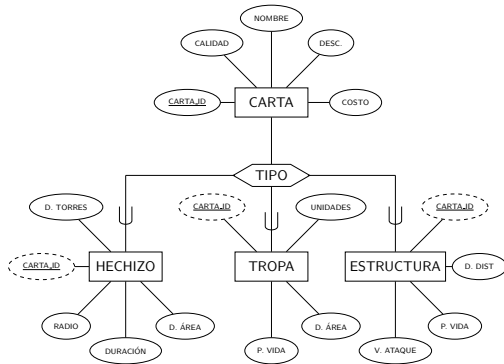
Tropa(#C, Unidades, P. Vida, D. Área)

FK: #C REFERENCES Carta

Estructura(#C, D. Dist, P. Vida, V. Ataque)

FK: #C REFERENCES Carta

Diseños para la especialización (partición)



Hechizo(#C, Nombre, Calidad, Desc., Costo, D.Torres, Radio, Duración, D. Área)

Tropa(#C, Nombre, Calidad, Desc., Costo, Unidades, P. Vida, D. Área)

Estructura(#C, Nombre, Calidad, Desc., Costo, D. Dist, P. Vida, V. Ataque)

Diseños para la especialización (partición)

Carta(#C, Nombre, Calidad, Desc., Costo)

Hechizo(#C, D.Torres, Radio, Duración, D. Área)

FK: #C REFERENCES Carta

Tropa(#C, Unidades, P. Vida, D. Área)

FK: #C REFERENCES Carta

Estructura(#C, D. Dist, P. Vida, V. Ataque)

FK: #C REFERENCES Carta

Hechizo(#C, Nombre, Calidad, Desc., Costo,
D.Torres, Radio, Duración, D. Área)

Tropa(#C, Nombre, Calidad, Desc., Costo, Unidades,
P. Vida, D. Área)

Estructura(#C, Nombre, Calidad, Desc., Costo, D.
Dist, P. Vida, V. Ataque)

Diseños para la especialización (partición)

Carta(#C, Nombre, Calidad, Desc., Costo)

Hechizo(#C, D.Torres, Radio, Duración, D. Área)

FK: #C REFERENCES Carta

Tropa(#C, Unidades, P. Vida, D. Área)

FK: #C REFERENCES Carta

Estructura(#C, D. Dist, P. Vida, V. Ataque)

FK: #C REFERENCES Carta

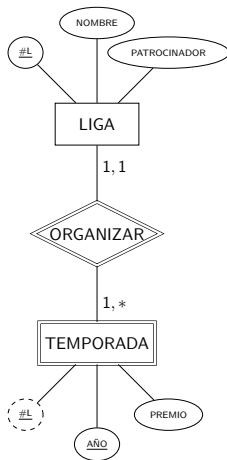
Hechizo(#C, Nombre, Calidad, Desc., Costo,
D.Torres, Radio, Duración, D. Área)

Tropa(#C, Nombre, Calidad, Desc., Costo, Unidades,
P. Vida, D. Área)

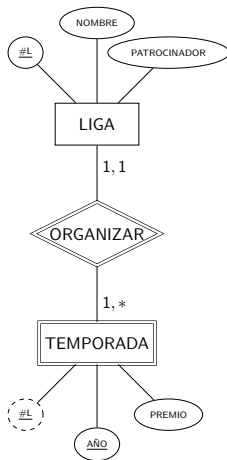
Estructura(#C, Nombre, Calidad, Desc., Costo, D.
Dist, P. Vida, V. Ataque)

TAREA: ¿Cuándo uno es mejor que el otro?

Diseño para las entidades débiles



Diseño para las entidades débiles

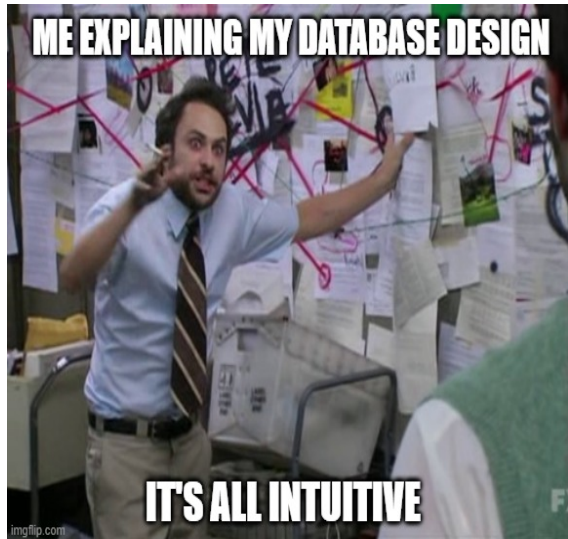


Liga(#L, Nombre, Patrocinador)

Temporada(#L, Año, Premio)

FK: #L REFERENCES Liga

El diseño lógico no es subjetivo



Entonces...

... alguna duda?



Bases de Datos

Diseño Lógico: Modelo relacional. Diseño Intuitivo

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

6 de febrero de 2024

Anexos

Condiciones

1. Dos relaciones R y S , no necesariamente distintas.
2. R y S tienen el mismo esquema, exceptuando, quizás, el nombre.

Operaciones de conjuntos

- ▶ $R \cup S$ es una relación con el mismo esquema, a excepción del nombre, cuyo cuerpo consiste en las tuplas que pertenecen a la relación R o a la relación S o ambas. Las tuplas duplicadas se eliminan.
- ▶ $R \cap S$ es una relación con el mismo esquema, a excepción del nombre, cuyo cuerpo consiste en las tuplas que pertenecen tanto a la relación R como a la relación S .
- ▶ $R - S$ es una relación con el mismo esquema, a excepción del nombre, cuyo cuerpo consiste en las tuplas que pertenecen a la relación R y no a la relación S .

Anexos

Operaciones que remueven parte de una relación

Restricción o Selección ($R \sigma F$)

$R \sigma F$ produce una nueva relación con el mismo esquema que R , a excepción del nombre, cuyo cuerpo es un subconjunto del cuerpo de R . Es decir, las tuplas en la relación resultante son aquellas tuplas de R que satisfacen la condición F , expresada mediante una fórmula bien formada.

Anexos

Operaciones que remueven parte de una relación

Proyección ($\pi_{A_1, A_2, \dots, A_n}(R)$)

$\pi_{A_1, A_2, \dots, A_n}(R)$ produce una relación cuyo esquema solo contiene los atributos A_1, A_2, \dots, A_n de R . El cuerpo de la relación consiste en todas las tuplas $\{(A_1 : a_1), (A_2 : a_2), \dots, (A_n : a_n)\}$ tal que existe una tupla en R cuyo valor asociado al atributo A_i es a_i para todo $i = 1, \dots, n$.

Producto cartesiano ($R \times S$)

Es una nueva relación cuyo encabezado es la unión de los encabezados de la relación R y la relación S . La llave primaria de la nueva relación es la unión de las llaves primarias de R y S . Y las llaves foráneas de R y S también son llaves foráneas en $R \times S$. El cuerpo consiste en el conjunto resultante de unir cada tupla de R con cada una de las tuplas de S .