

BASES DE DATOS

(• • •)

* En el DNI no es necesario con la fecha de nacimiento

D M A

M A

Bonita

Septiembre 23 de 2021

Introducción a Bases de datos

- La disponibilidad de información y gestión de manera eficiente es fundamental
- **Atributos:** Metadatos que tienen valores asociados
- Cualquier representación correlacionada puede ser definida como dato, que al ser interpretada me proporciona información.
- Los datos me trae información

- Sistemas Manejadores de BD.

Un DBMS es una colección de datos interrelacionados y un conjunto de programas para acceder a ellos.

También es capaz de administrar grandes colecciones de datos que son compatibles y persistentes, garantizando consistencia y privacidad.

• **BD = SMBD**

Modelo	Jerárquico	Relacional	Relacional	Documental
Leng. Consulta:	leng. lms	SQL	NoSQL	NoSQL

Septiembre 27 de 2021

Continuación..

- Todo SMBD tiene un:

- Modelo
- Lenguaje de consulta

- Archivos vs SMBD

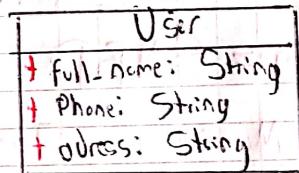
- Un sistema de archivos es una abstracción del CRUD
- Los archivos pueden presentar redundancia de datos e inconsistencias (-)
- Pueden existir diferentes versiones (-)
- El SMBD tiene independencia de los datos (Es independiente lo que esté en disco de la visualización)
- El lo guarda en archivos pero de una manera estructurada
- Proporciona mecanismos para el CRUD de manera eficiente.
- Optimiza consultas, soporta archivos de gran tamaño.

- Se reduce el tiempo de desarrollo, debido a que tienen control de concurrencia, recuperación de errores, etc.
 - Independiente de datos → Física y Lógica

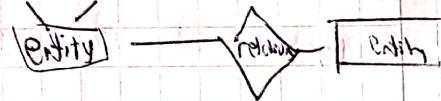
- Modo de datos

- Es una colección de herramientas semánticas de datos y restricciones conceptuales para describir datos, relaciones de datos.
 - Descripción: ¿Qué voy a gestionar?
 - Relaciones: ¿Qué relaciones existen?
 - Semántica: ¿Significado de las cosas?
 - Restricciones: ¿Qué no puedo poner? Para que no pierda de sentido o coherencia

1) **Modelo relacional**: Colección de tablas con nombre único.



2) **Modelo entidades-relaciones:** Colección de entidades que representan objetos del mundo real y sus relaciones



3) Modelo de datos semi-estructurado

4) Modo de datos buscado en objetos

- Cada fila puede considerarse como un "registro" [Ej.] (Es un elemento de conjunto)

- Funciones SMBD

- Control de concurrencia
 - Seguridad
 - Recuperación - backups
 - Integridad
 - Descripción de los datos

Hoy a Ti Físico y Lógico

- Modelo relacional

- Cada relación es descrita usando un esquema (Schema)

Ej.: Estudiante (nombre: char,
codigo: char,
programas: char) } (fisicamente de estudiante)
} tiene un nombre y un listado de atributos
de relación en su respectivo dominio

- Para definir un esquema necesitamos un lenguaje de definición de datos (DDL)

Ej.: CREATE TABLE <> Nombre <>

ALTER TABLE <> Nombre <>

- Existe el esquema físico que me guarda mis datos, como serán guardados

Septiembre 30 del 2021

Contenedores

- Uso recursos del Sistema Base
- Progress se instala dentro de un Contenedor

- Virtualización

- El Hypervisor es un proceso que crea y ejecuta máquinas virtuales

• Mezcla "resource pooling", escalabilidad, etc.

- Es posible tener VM's en la nube

• Alia así tiene limitaciones

• Requieres CPU, Almacenamiento, RAM,

• La portabilidad no está asegurada

- ¿Contenedores? (Docker)

- Deben de tener nombre único

• Un docker puede volverse una "plantilla" (Image)

- Comandos Docker

• docker ps, docker ps -a

• Obtener imagen de postgres: Buscar postgres en docker hub



"mondo_db2"

Scribe

(Ctrl + Shift + R)

alt Gr + ; ,

Lenguaje =

SQL

- docker pull postgres

• 'rm -frv proyecto_db/' → Borrar proyecto_db

• git clone 'repo aquí'

• ls para verificar.

• Cd proyecto_db/

• Se abre la carpeta en Visual Studio.

- Mos info

• El archivo Dockerfile se encarga de crear la Base de datos al arrancar

• \$ Cd database_postgres

• docker Step (First 4 letters id) → Poner

• docker rm → Eliminar

• Tutorial: undCortijo / proyecto_db

• docker exec -it (Conectar a la máquina de docker)

• docker exec -it 9755 /bin/bash

↳ docker exec -it 9755 /bin/bash
TJ del docker

• psql postgres postgres

• \l → List of database

• \c mundo_db

↳ A partir de obs. ya puedo entrar Comandos de SQL

Octubre 4 de 2022

Scribo

¿Qué es el SGBD?

- Sistema de Gestión de Base de datos. Sirve para crear y manejar una base de datos.

- Puedes tener 1 o más BD

• ¿Por qué utilizarlo?

- Acceso eficiente e independencia

- Reducción de tiempo de desarrollo

- Acceso concurrente

- Seguridad de datos

- Recuperación de fallos

- Todo gestor tiene: Modelo de datos y un lenguaje de consulta.

- Vista: Como el DB ve los datos

- Esquema conceptual: Define estructura lógica (filas, columnas, Son relaciones) Atributos

- Esquema Físico: Describe ficheros e índices utilizados. Características del individuo

- Componentes

- El SGBD siempre guarda en disco, y manipula datos.

- El motor de Base de datos (DBMS): Procesador de consultas, Gestor de Transacciones y gestor de almacenamiento. ①

- Árbitro de las transacciones, quien vigila las consultas. ②

- E\$

↓
Módulos
Componentes

- Procesador de consultas

- DDL: lenguaje de definición. (Interpret) \rightarrow CFBID

- DML: lenguaje que modifica (Computador)

- Motor de evaluación.

- Gestor de almacenamiento

- Gestor de transacciones: Asegura las BD ante fallas

-

-

$\rightarrow \rightarrow$ Gestor de transacciones

- Garantiza ejecución de transacciones

- $\circ - o - o - o - o - o - o - o$

Diseño Conceptual "Lo que yo pienso"



Esquema Conceptual

E-R (Entidad-Relación)



Diseño lógico



Esquema lógico REL



}} Causas encargadas de la BD

- Funciones

- ① Personas del DBA:
- ② Usuarios espontáneos
- ③ Yo tengo programado que me llegue lista la información
- ④ Aplicaciones en general, usuarios de las aplicaciones.

- Abstracción de datos
- Independencia de datos
- Consistencia de datos, Sin redundancia

- Lenguajes

Estructuras:

- DDL (Data Definition L)
- STDL (Storage Definition L)
- VDL (View definition L)

Manipulación:

- DML: Modificaciones de los instancias
- QL, Subconjunto del DML

MODELING DATA

- MER

- The entity-relationship model is a conceptual data model

It has graphical notation ("Tables")

(Model conceptual)

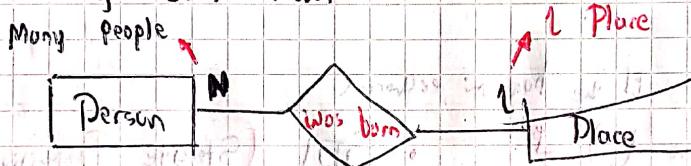
- = Entity - Group (Conjunto) → It has to be tangible



 = Attribute

 = Relationship - Relation (It has a verb)

- We are using Chen model



- Data requirements are described in terms of the conceptual schema

- Be careful with singular and plural (e.g.)

- Instance: An element from an entity ("A car with plate AAA-111")

- Attributes

1. Simple " " → Only one "thing"

2. Composite " " → An attribute that has attributes.

3. Single Valued " " → Simple and only one value (e.g., Age is numeric)

4. Multivalued " " → More than one "things" →

5. Derived " " or Structured " " → It can extract that Attribute from other

6. Key " " → Identifies an instance because it's unique



(Age)
(Birthdate)

ID

326531864321

- Relationship

- It doesn't have to be a verb

- \diamond only joins \square

Octubre 8 de 2022

- Posibles preguntas Q&A

- ¿Qué es un esquema? $\text{E} \rightarrow$ Esquema lógico
Esquema física
Esquema conceptual

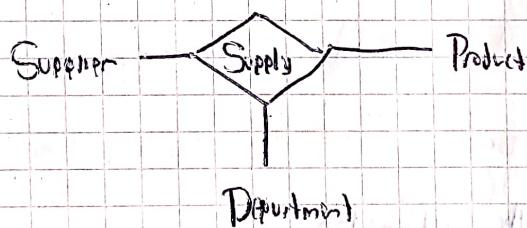
• La lista de atributos con su dominio.

Continuación...

- Relationship



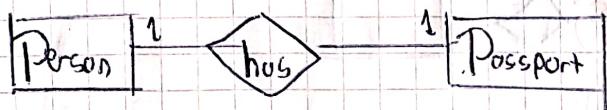
- It's possible to have "ternary", but everyone has to be completed with the \diamond



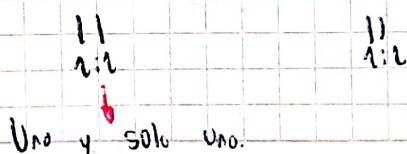
Cardinality

- One-to-one

- When only one instance of an entity is associated with the relationship, it's marked as 1:1.



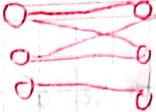
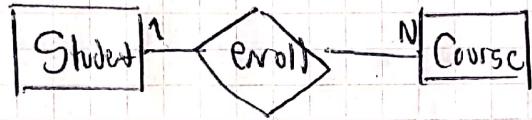
- Crows-Foot



Uno y solo uno.

Scribe

- One to many



"Student made inscriptions 1 to N Courses"

1:1

1:N

0:N

K

- Many to one

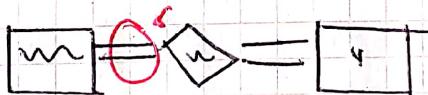
• The same of above but backwards

- Many-to-many

- Participation

- Total participation

• Each entity is involved in the relationship



- Partial participation



Ejercicio - Anotaciones

- Locutions → Plurul → It has to be an entity

Octubre 11 de 2022

C. Continuación...

- Review

- Entidad - Relación (MER-ER)

• Entidad

• Relaciones: vínculos entre 1 o más.

• Atributos: características que tiene una entidad

• Abstracción del mundo real

• Todo es conceptual → Esquema Conceptual (Diagrama)

- Notación

• Chen (1976)

• Entidades: \square

• Entres: $-$, $=$, $---$

• Cardinalidad: 1-1, 1-N, M-N

• Atributos: Simple \circ , Objeto \odot , \odot , \odot , Solo de muchos a muchos puedo usar letros diferentes en Mayúscula

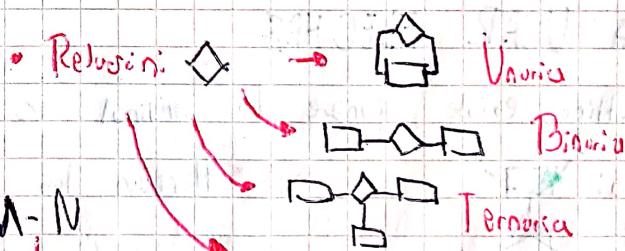
• Crow Foot (1976)

• Entidades: Igual

• Entres: —

• Cardinalidad: $\square \rightarrow 0$, $\square \rightarrow 1$, $\square \leftrightarrow 1:1$, $\square \oplus 1$

• Atributos: \circ , \odot , \odot , \odot , \odot

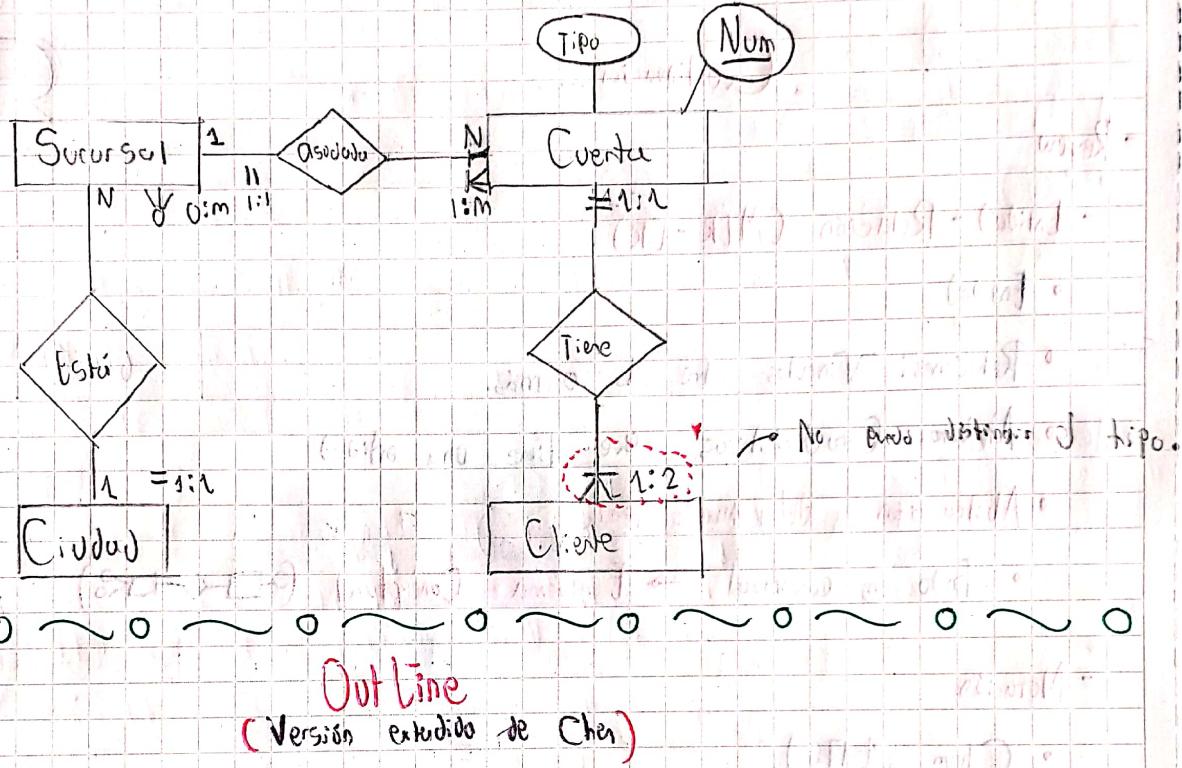


Scribe

- ¿Qué hacen en un ejercicio?

1. Identificar entidades

2. Identificar Relaciones (Verbos)



• NUNCA USAR FLECHAS

• El Outline existe porque la original es muy simplista

• Herencia: 

- EER

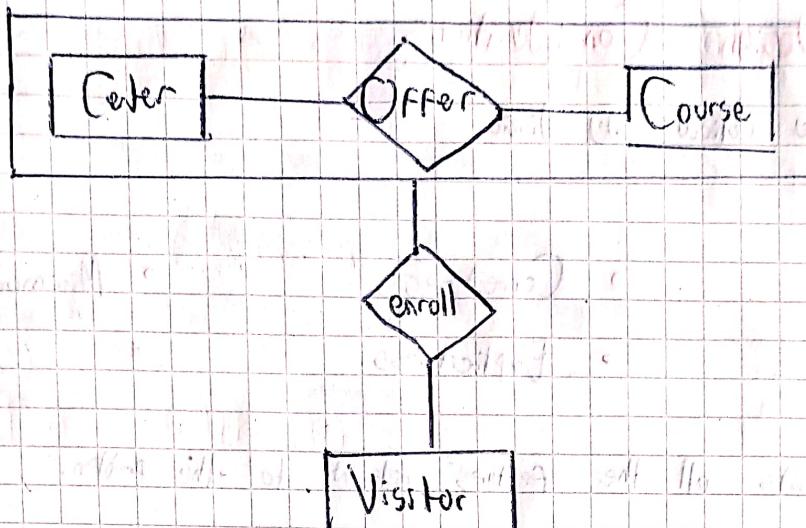
Se incluye:

• Specialization → An entity is sub-divided in sub-entities.

• Generalization → "A person is student or teacher"

• Aggregation → T. con tie " ◊---◊ "

- "Centro que ofrece un curso, un visitante se va a matricular"



- Ejercicio

Otoño 18 de 2022

Conceptos: Entidades

- Orden

1. Entidades

2. Relaciones

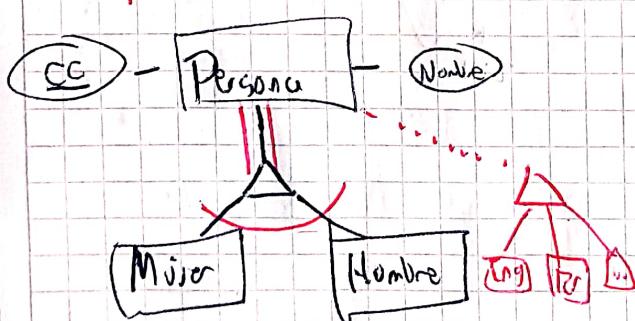
3. Atributos

4. Cardinalidad

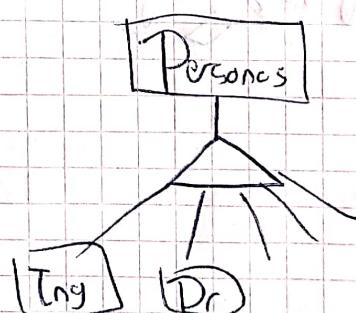
- G y E

No hay continuidad

No es posible II



Generalización



Especialización.

II = Obliga generalización (Participación total), (I) Partiu (Parcial)

↙ = Prohibido solapamiento. (Exclusivo) ↗ = Solapeando

△ = Un mujer y un hombre pueden ser doctores, ingenieros, etc.

Design Considerations

- Se intenta evaluar la calidad de diseño.

- Métricos:

- Completeness
 - Correctness
 - Minimal factor
 - Expressiveness
 - Explicitness.
- Completeness: It represents all the features relevant to the problem.
 - It has all the concepts.
 - All requirements are achievable

- Correctness: When it's correct Syntactically and Semantically.
 "I used only one" "It's consistent with the problem"

- Minimal factor: It's being repeated OK

- Expressiveness: It represents reality in a natural way
 "That other person can rebuild the problem only reading my diagram".

- Explicit: It uses no formalisms other than the ones from the language.



\Leftarrow ✓
Z:N

D M A
Octubre 21 del 2022

MODELO RELACIONAL

- Implementado en 1980 en varios motores (DB2, Oracle, SQL, MySQL, PostgreSQL)
 - Los entidades son representados en **tublos**
 - Los filos representan los elementos en el conjunto
 - Los columnas representan atributos.
 - **Instancia:** Un individuo de un conjunto (dato)

The diagram illustrates a 3D cube with three dimensions:

- CARDINALIDAD** (Cardinality) is represented by the vertical axis.
- GRADO** (Degree) is represented by the horizontal axis.
- Atributos** (Attributes) is represented by the depth axis.

The cube is divided into four main regions:

- (Piso por piso)** (Floor by floor) is the top layer.
- (Nombre)** (Name) is the middle layer.
- (Frúndimiento)** (Imprinting) is the bottom layer.
- Tuplas** (Tuples) is located at the bottom right corner, connected by red arrows from the depth axis and the bottom layer.

Red lines and arrows highlight the cardinality dimension, showing how it applies to all three axes.

- Cada tupla está asociada con un atributo
 - Cada atributo tiene un dominio (**D**: Colono, D: Niño/a, D: Fecha)
 - El esquema se utiliza para describir una relación R

Nom de $\rightarrow \text{R}(\text{A1:D1}, \text{A2:D2}, \text{A3:D3})$

En resumen, el Síndrome de estreñimiento constante es la relación entre la infusión y los cambios.

- La relación 'r' de estructura 'R' es un conjunto de n-tuplas (-)

$$r = \{t_1, t_2, t_3, \dots, t_n\}$$

↳ # (Cardinality)

- In relation $r(\pi)$ true $\text{dom}(A_1), \text{dom}(A_2), \dots$

DEFINICIÓN DE LOS DATOS

- Attributos llave

- No se repite
- Si hay más de un conditivo o llave primaria, one of them is chosen by the DBA. It can't have null
- Foreign Key: A key from other entity-table. It relates a tuple with another tuple in another table.

LENGUAJE DE DEFINICIÓN DE DATOS

- DDL

- Permite especificar la estructura de la base de datos junto con sus restricciones.
- DDL = SQL (En este curso)

• EJ: CREATE TABLE

```
id Person INT(11) UNIQUE NOT NULL  
Person_name CHAR(60)
```

• = Attribut

• = Único

• = Restricciones

- Restricciones de llave primaria

- Cada relación (talla) debe de tener una llave primaria.
- Hay más.

- **Claves primarias:** No puede ser null, debe ser única y son indexadas automáticamente.
- "PRIMARY KEY" → Ya viene cargado "UNIQUE NOT NULL"

• Restricciones de clave primaria

→

- Verifico

Todos
Expedición: Date
NúmeroIdentidad: String
Num: String PK
tipo: String

Professor
ID: String PK
Nombre: String
Apellido: String

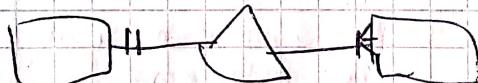
→ 'Sí' es al ser un
dominio, no posee

Cursos
ID: String PK
Creditos: Int

Ubicación
ID: String PK
Calle_direccion: String
Calle_Número: String
metros: Int

→ Este esquema incluye el
diagrama lógico

• Relationship



→ La clave primaria siempre va al muchos.

Octubre 25 de 2022

DEPENDENCIAS FUNCIONALES

- Ejercicio

Stuff
ID: String PK
Birth: Date
Phone: String
Address: String
Name: String

Company
ID: String PK*

Wife
Name: String
ID: String PK*

Child
Name: String

Task
Description: String
ID: String PK*

Phone
Number: String
Type: String
Owner: String *

- Todo entidad se convierte en una relación.
- Debido a que son entidades fuertes, deben tener un ID.
- Al ser multivalueado, se convierte en una relación.
- ① Me indica que es una llave fuerte

Task es un atributo de una relación, este se queda en el "Nodos".

Ejemplo:

Stuff (ID, Birth, Name, Address, id-Company*, Since)

Company (ID, name)

Wife (ID, id-Stuff*)

↑ Foreign Key

añadido para que un "Stuff" pueda tener muchos hijos.

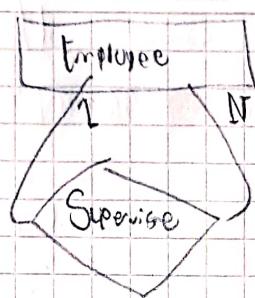
Child (Name, id-Stuff*, id)

Task (ID, description)

Phone era un atributo multivalueado, se convierte una relación.

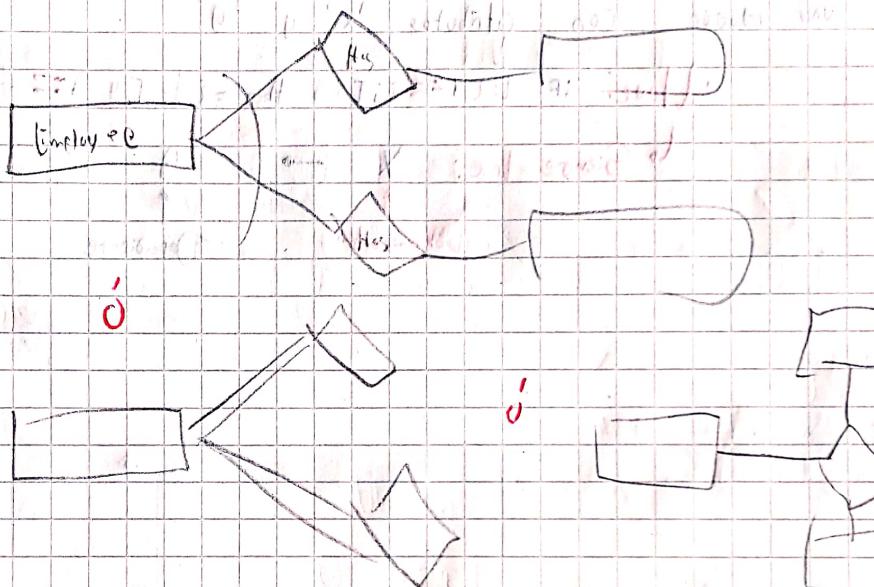
Phone (Number, type, owner*)

Task - Stuff (id-task*, id-stuff*) } otra relación debido a que es de muchos a muchos



\rightarrow Employee (ID, ID-Endorse*)

• Obligaciones:



- Otros:

Employee (email, phone, name, id)

Login (Username, password, id_employee*)

Department (name, description, id, id_employee*)

Role (id, name, description)

Employee - Role (id_employee*, id_Role*)

- Transforming a Generalization / Specialization

- Hay tres formas de resolverlo

DEPENDENCIAS FUNCIONALES

- Sea " R " una relación con atributos ' x ' y ' y '

• Movet: if $t_i[x] = t_j[x]$ then $\{t_i[y] \Rightarrow t_j[y]\}$

<u>R</u>	<u>1</u>	<u>1</u>
1	1	1
2	1	2
3	2	2
4	3	3
4	4	6

↳ Siempre true: $x \rightarrow y$

Determinante Dependiente

- Sea

$$R(A_1, A_2, A_3, \dots, A_n)$$

$$r(R) = \{ t_1, t_2, t_3, \dots, t_m \}$$

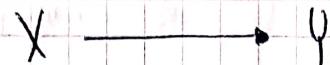
$$\text{↳ } t_{ij} = \langle V_1, V_2, V_3, \dots, V_n \rangle$$

- ¿Para qué los necesitamos?

- Para la Semántica de los atributos
- Para la Reducción de redundancia
- Reducción de valores null
- Prohibición de tuplas falsas (Spurious)

- ¿Qué es una dependencia funcional?

- Relación entre dos atributos, típicos entre la PK y non-PK en una misma tabla.



Determinante Dependiente

- Existe: Uvres primoros, Uvres úncos, Uvres furónos, Condicodos a llors.
- Condicoda o llore: Que per sus caraderisticos, puede ser una llore.

Octubre 28 del 2022

Mis Comandos - Práctica

- Mande -Jb ≠ # (d) b Vr folios

- CREATE TABLE

munde -Jb ~~#~~ → Entrar

- ALTER TABLE

- DROP TABLE

b;: Venados (iD int, nombre char(100)) .

Cliete(iD int, adir -C String, udir -# String, nombre char(100) , tel char(100)

Orden C

← Medicamento(Code Int, Cont int, lab char(100) , lote i

Medicamento

- Ver apuntes de Carlos

Noviembre 1 de 2022

Continuación - Dependencias Funcionales...

- Atributos primos: Que son candidatos a ser llaves
- Atributos no primos: Que no son primos Los atributos llaves son
- Características de las llaves: No NULL, No repetitivos, indexados automáticamente.
- Primary key - Alternate keys - Unique keys: Los tres me forman una Super Key.
 Que pueden ser una alternativa
 a la Primary key Valores únicos, de hecho permiten "NULL"
- Definitions: Functional dependency
- "Definición Formal": $\forall t_1, t_2 \mid t_1[A, B] = t_2[A, B] \rightarrow t_1[D] = t_2[D]$
- Trivios:

$$A \rightarrow A$$

$$B \rightarrow B$$

$$C \rightarrow C$$

$$D \rightarrow D$$

- Dependencias funcionales que se matemáticamente se refieren a $R(A_1, A_2, A_3)$

$$D = \{ A_1 A_2 \rightarrow A_3 \\ A_2 \{ A_2 A_3 \rightarrow A_2 \\ \quad A_2 \rightarrow A_3 \\ \quad A_2 \rightarrow A_2 \}$$

Puedo afirmar que A_2 es una candidata a llave

• Keys

- It's an attribute or a combination of attributes that uniquely identifies each row within a table.
- It has uniqueness
- Ejemplo ejercicio:

$$R(A, B, C, D, E)$$

$$A \rightarrow D$$

$$D \rightarrow B$$

$$B \rightarrow C$$

$$B \rightarrow E$$

$$\text{Propósito: } A \rightarrow B, \quad B \rightarrow C, \quad A \rightarrow C$$

FD: :: A (A is candidate a func)

$\alpha \sim \alpha \sim \alpha \sim \alpha \sim \alpha$

- Armstrong's axioms

- It's used to infer all the functional dependencies on a relational database.

DBA: Database administrator.

- Axiom of reflexivity

- If ' α ' is a subset of ' β ', then ' β ' determines ' α '

• Ex: $A, B, C \rightarrow A$

$A, B, C \rightarrow B$

$A, B, C \rightarrow C$

$A, B, C \rightarrow A, B$

- Axiom of augmentation (Partial dependency)

- If ' β ' determines ' α ', then ' $\beta\Omega$ ' determines ' $\alpha\Omega$ ' for any ' Ω '

• Ex: $A \rightarrow B$

$A, D \rightarrow B, D$

$A, B, C \rightarrow B, C$

- Axiom of transitivity

- If ' α ' determines ' β ', and ' β ' determines ' Ω ', then ' α ' must also determine ' Ω '.

• Ex: $A \rightarrow B$

$C \rightarrow B$

$B \rightarrow C$

$A \rightarrow C$; per transitiv

$A \rightarrow E$; "

- Axiom of Union

- If ' β ' determines ' α ' and ' β ' determines ' Ω ', then ' β ' determines ' $\alpha\Omega$ '

• Ex: (Dej anterior)

$A \rightarrow CBE$; Por Union

Axiom of decomposition

- If ' B ' determines ' α_Ω ' then ' B ' determines ' α_C ' and ' B ' determines ' Ω'

• Exist (Def anterior)

• $B \rightarrow D$; decomposing

$B \rightarrow E$

$B \rightarrow C$

$CD \rightarrow D$

$D \rightarrow D$

$C \rightarrow D$

- Exercise

$$\bullet R = \{A, B, C, D, E, F\}$$

$$D = \{A \rightarrow B, A \rightarrow C, CD \rightarrow D, CD \rightarrow F, B \rightarrow E\}$$

$$\textcircled{1} \quad A \rightarrow E.$$

$$\bullet B \rightarrow E$$

$$\bullet A \rightarrow E$$

$$\textcircled{2} \quad A \rightarrow F$$

$$\textcircled{3} \quad A \rightarrow C$$

$$\textcircled{2} \quad C \rightarrow D \text{ Decomposing } (CD \rightarrow D \text{ solo por trivios})$$

$$\textcircled{3} \quad A \rightarrow D; \text{ Transitivity}$$

$$\textcircled{4} \quad A \rightarrow CD; \text{ Unión}$$

$$\textcircled{5} \quad CD \rightarrow F; \text{ Transitivity}$$

$$6. \quad A \rightarrow F$$

$$\bullet R(A, B, M, E, F, N) \text{ such that } M \rightarrow EF$$

$$D = E \quad M \rightarrow N$$

$$M \rightarrow A$$

$$M \rightarrow B$$

$$AB \rightarrow EF$$

$$\bullet M \rightarrow AB \quad \text{Unión}$$

$$\bullet M \rightarrow EF \quad \text{Transitivity}$$

Noviembre 4 de 2017

Scribe

Continuación..

- Anomalías

- Redundancia: Valores que se repiten

- Anom. Actualización: Inconsistencias que se generan después de una actualización.

- Anom. Eliminación: Pérdida no intencional de valores

- Anom. Interción: Imposibilidad de adicionar valores o tuplos

- Ejercicios

- Pseudofuncionalidad

- $\{x \rightarrow y \wedge yw \rightarrow z\} \rightarrow Xw \rightarrow z$

- Redundancia

- $y \subseteq x \rightarrow x \rightarrow y$

- Aumento

-
-
-
-
-

- Ejercicio 2

- C, para $B \rightarrow D$

- Ax, para $E \rightarrow B$

- B √, D √

- Exercise 3

$R(A, B, C, G, H, I)$

$$D = \{ \begin{array}{l} A \rightarrow B \quad 1 \\ A \rightarrow C \quad 2 \\ CG \rightarrow H \quad 3 \\ CG \rightarrow I \quad 4 \\ B \rightarrow H \quad 5 \end{array}$$

- $CG \rightarrow HI$ (Por Unión de ③ y ④)

- $AG \rightarrow I$ (Pseudotransición de ② y ④)

- $A \rightarrow I$ X

- $B \rightarrow I$ X

Definición

• Sea F un conjunto de TDF

• El Cierre de F (F^+) es el conjunto

- Exercise

$F = \{M \rightarrow N, M \rightarrow A, M \rightarrow B, AB \rightarrow F\}$ Debejo solo si me dan el conjunto 'F'

$R(A, E, TS, F, M, N)$

- Probar $M \rightarrow EF$

$$\begin{array}{ll} \cdot M \rightarrow A & \text{Unión} \\ M \rightarrow B & M \rightarrow AB \\ AB \rightarrow EF & \text{Truncamiento} \\ M \rightarrow EF & \end{array}$$

- Exercise 3

• Cuales dependencias satisfac.

• $BE \rightarrow D$ ✓

• $C \rightarrow AB$ X

• $B \rightarrow D$ X

• $AD \rightarrow E$ ✓

• $E \rightarrow B$ X

D M A

Scribo

- Hallar el recubrimiento mínimo

$$D = \{ A \rightarrow BC, AB \rightarrow C, AC \rightarrow D, B \rightarrow C, A \rightarrow B \}$$

- Paso 1:

• Hallar D'

$$D' = \{ A \rightarrow B, A \rightarrow C, B \rightarrow C, A \rightarrow B, AB \rightarrow C, AC \rightarrow D \}$$

- Paso 2:

• Eliminar atributos redundantes

$$AB \rightarrow C \quad \{ B^+ = \{B, C\} \} \Rightarrow D' = \{ A \rightarrow B, A \rightarrow C, B \rightarrow C, B \rightarrow C, AC \rightarrow D \}$$

Hacía donde tiene que determinar

• $AC \rightarrow D \quad \{ C^+ = \{C\} \}$

$$\{ A^+ = \{A, B, C, D\} \}$$

$$D = \{ A \rightarrow B,$$

$$A \rightarrow C,$$

$$B \rightarrow C,$$

$$A \rightarrow D \}$$

- Paso 3:

• Eliminar dependencias redundantes

$$D' - \{ A \rightarrow B \}; A^+ = \{A, C, D\} \therefore \text{No sobra}$$

$$D' - \{ A \rightarrow C \}; A^+ = \{A, B, D, C\} \therefore \text{Sobra}$$

$$D' - \{ B \rightarrow C \}; B^+ = \{B\} \therefore \text{No sobra}$$

$$D' - \{ A \rightarrow D \}; A^+ = \{A, B, C\} \therefore \text{No sobra}$$

$$\therefore D' = \{ A \rightarrow B, B \rightarrow C, A \rightarrow D \}$$

NORMALIZATION

- Pasee de orquesta sus dutos en uno BD. Esto incluye crew y sus
relojadores.

- Normal Forms

- 1 - NF

- Todos los atributos deben tener valores simples. No puede haber un atributo dentro de él mismo con múltiples dominios. Se espera un valor único.

November 1 of 2022

Continuation...

- Alímara: Que los datos del dominio son invisibles.
 - Todos los atributos deben ser atómicos

- 2 - NF

- "Todo atributo no primo _ deve depender de_ un atributo primo (llave) ".

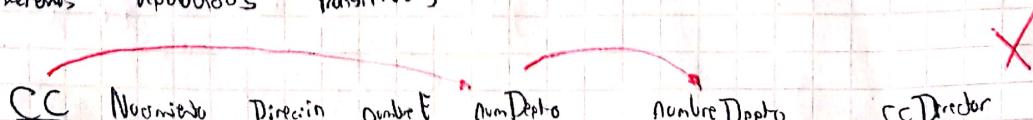
• ID NumProg hours numEmployee ~~X~~ numProg ✓ OfficeProg

numbered attribute violates 2NF

- Los atributos primos se forman todos en Conjunto.
 - "Los atributos no primos dependen de todos los atributos primos en conjunto".
 - También para que esté en 2^{-NF}, debe de estar en 1-NF

- 3 -NF

- Debe de estar en 2-NF, y por lo tanto, 1-NF
 - "No queremos dependencias transitivas"



Solução: CC Número Diretório número num Dots

num Depto nombre Depto cc Director

- Si no existe ninguna dependencia funcional transitiva.

- 4 - NF

- * "Cada tributo debe de tener asignado un solo valor"

جتنی نجات مدنی

CUCENTE

CD	Clave	Nombre	Apellido	Teléfono
~	~	~	~	~
4SG		James	Wright	4864113 4863212
~	~	~	~	3734422

- Se lleva a la NF de lo siguiente mues:

ID 4 SG	Nombre	Apellido	ID 4 SG	Teléfono 4 SG

$\text{O} \sim \text{O} \sim \text{O}$
Dependencia parcial.

DIRECCIÓN CORREO SUST CRIPTOR

ID Dirección del correo Primer Nombre - Apellido
752 Card @~
752 Crookston @~

• Hallar la cerradura (F') $R(A, C, D, E, F)$

$F = \{ A \rightarrow B, A \rightarrow C, CD \rightarrow E, CD \rightarrow F, B \rightarrow E \}$ • Aumento

• $A^+ = \{ A, B, C, E \}$ → $A \rightarrow E$ • Transivida

• $D^+ = \{ B, E \}$ • Treflexiva

• $CD \rightarrow F$ ^ $CD \rightarrow E$ → $CD \rightarrow F$ • Unión

∴ $F' = \{ A \rightarrow BC, A \rightarrow E, CD \rightarrow FE, B \rightarrow E \}$

• Hallar Reclamante mínimo

$A \rightarrow BC$ $B \rightarrow C$ $A \rightarrow B$ $AB \rightarrow C$

1. Descompongo

$D' = \{ \underbrace{A \rightarrow B}_{I}, \underbrace{A \rightarrow C}_{II}, \underbrace{B \rightarrow C}_{III}, \underbrace{A \rightarrow B}_{IV}, \underbrace{AB \rightarrow C}_{V} \}$

2. Elimino dependencias redundantes (?)

$A^+ = \{ A, B, C \}$ $AB \rightarrow C \quad \left\{ \begin{array}{l} A^+ = \{ A, B, C \}; A \rightarrow C \\ B^+ = \{ B, C \}; B \rightarrow C \end{array} \right.$ (mejoró con una o con la otra)

$B^+ = \{ B, C \}$

3.

$D' - I ; A^+ = \{ A, C \}$ ✓

$D' - II ; A^+ = \{ A, B, C \} \times \rightarrow$ Reclamante

$D' - III ; B^+ = \{ B, C \}$ ✓

$D' - IV ; A^+ = \{ B \}$

~(B → D)

- $F = \{AB \rightarrow C, C \rightarrow D, B \rightarrow C, C \rightarrow D, \neg B \rightarrow D\}$

- $\neg B \rightarrow D \checkmark$

Noviembre 15 del 2022

Ejercicios

$$\begin{array}{c} \bullet YX \rightarrow Y \text{ (Comut)} \\ \uparrow \\ \bullet WY \rightarrow Y \\ \text{(P. Transitivity)} \end{array}$$

a) $\{W \rightarrow Y, X \rightarrow Z\} \models \{WX \rightarrow YZ\}$

- No hay mano
- $WX \rightarrow YZ$
- $WX \rightarrow Y$

b) $\{X \rightarrow Y, Y \rightarrow W, WY \rightarrow Z\} \models \{X \rightarrow Z\}$

- $X \rightarrow WY$ (Unión) (I, II)

- $X \rightarrow Z$ (Transitividad) (III)

c) $\{XY \rightarrow Z, Z \rightarrow X\} \models \{Z \rightarrow Y\}$

- $XY \rightarrow X$
- $Z \rightarrow X$
- $ZY \rightarrow Z$

- $Y \rightarrow Y$
- $ZY \rightarrow XY$

- $Y \rightarrow Y$
- $Z \rightarrow XY$

$$\begin{matrix} X & Y & Z \\ X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \end{matrix}$$

$Z \rightarrow Y$ no es una dependencia
funcional de ese conjunto

I II

d) $\{X \rightarrow Y, XY \rightarrow Z\} \models \{X \rightarrow Z\}$

- $\{X \rightarrow Y, XY \rightarrow Z\} \rightarrow XX \rightarrow Z$ (Producto)

- $XX \rightarrow XX$ (Descomposición)
- $XX \rightarrow X$

- $X \rightarrow Z$ (Unión)

- III $X \rightarrow X$ (Trivial)

- IV $X \rightarrow XY$ (Unión entre I y III)

- V $X \rightarrow Z$ (Transitividad II y IV)

No resuelto

Nota:

$$\cdot \{ \alpha \sqsupseteq \beta \} \models \alpha \rightarrow \beta$$

$$\{ \alpha \subseteq \beta \} \models \beta \rightarrow \alpha$$

a) $\{ x \rightarrow y, y \sqsupseteq z \} \models \{ x \rightarrow z \}$

$$\cdot = \{ x \rightarrow y, y \rightarrow z \} \models \{ x \rightarrow z \}$$

$$\cdot$$

• Ver libro de Pérez. Fundamentos de Sistemas de Bases de datos. Ahí están los ejercicios.

ALGEBRA RELACIONAL

- It's the basic set of operations for the relational model
- The algebra operators thus produce new relations.
- **Operations:** Selection, Projection, renaming → "Unary operations"
They operate on a single relation.

Union, Difference, Cartesian Product →

~ Select

$$\sigma_{\text{condition}} (\langle R \rangle)$$

• **< condition >**: A boolean expression specified on the attributes of Relation R

• Tuples that make the condition true are Selected appear in the result of the operation

• " " " " " false are filtered out - discarded.

$$\cdot \text{Ej: Director} = \sigma_{\text{apellidos} = \text{"Director"},} (\text{Empleados})$$

$$\sigma_{\text{apellidos} = \text{GIL OR apellido} = \text{PEY}} (\text{Empleados})$$

O

$\langle \text{Comision} = \text{NULL} \& \& \text{Año-nac} < 70/01/01 \rangle$ (Empleado)

- Presentación "Proyección"

- $\Pi_{\text{atributos}} (\langle R \rangle)$

- $\langle \text{atributos} \rangle$: It's the desired list of attributes from relation R

- Creates a vertical partitioning. It keeps certain columns (Attributes).

- E.g.: $\text{Apellidos} = \Pi_{\text{apellidos}} (\text{Empleado}) \rightarrow$ It forms the column "apellidos".

- $= \Pi_{\text{apellidos}, \text{emp-no}} (\text{Empleado}) \rightarrow$ It takes two columns.

- It removes duplicate tuples - (Oficio, Director) ✓

- (Oficio, Director) X (It doesn't add it)

- It's not commutative.

- Renombrar

- $P_S \langle A_1, A_2, A_3, \dots, A_n \rangle (\langle R \rangle)$

List of attributes to rename.

- It renames the attributes of a relation or relation name.

o

- E.g.: $P_{T_1} = P_{\text{Directores}} (\text{Empleado}) \rightarrow$ Poso la tabla de empleados a Directores

Atribuc

P_{T_2}

Num-nero, apellido, ..., sueldo, ...

Para combinar
atributos, lista
atributos.

o Nombre de los
atributos los

Renombrar T_3 - $\sqrt{\text{Salario} = \text{Sueldo}} (\text{Empleado})$

→ Union

• $R \cup S$ 

Must be "type compatible"



• Duplicate tuples are eliminated

• Tuples are $\{a_1, a_2, b_1\}$ are compatible.• • $R \cup S$

A	B
a	1
a	2
b	1
b	3

 π_A

A	B
a	1
b	1

 π_B

A	B
a	2
b	3

• Intersection

• $R \cap S$ 

• Deben de tener igual dominio • Compatible

- Difference

• $R - S$ 

• Deben de tener esquema igual • y/o compatible

• Cartesian Product

• $R \times S$

$$\begin{array}{c}
 A \cup B \\
 X \{x_1, x_2, x_3\} \rightarrow A \times B \\
 Y \{y_1, y_2, y_3\} \\
 Z \{z_1, z_2, z_3\}
 \end{array}$$

rJoin

- It supports completely in the Cartesian Product

- It combines related tuples from various relations

$$R \bowtie S \quad \begin{matrix} \sim & \sim & \sim & \sim \\ \sim & \bowtie & \sim & \sim \\ \sim & \sim & \sim & \sim \end{matrix}$$

- Debe haber un atributo común en R y S

- Left Join: $R \bowtie S$

- Right Join: $R \bowtie S$

- Left-Right Join: $R \bowtie S$

- Aritméticas Generales: $\sim \sim \sim \sim \sim \sim \sim \sim$

• G Sum (Súmatoria) (Empleando)

$0 \sim 0 \sim 0$
Número 22 de 2022

CÁLCULO RELACIONAL

- Pensado para hacer consultas, pero el cálculo es totalmente declarativo mientras que el álgebra es procedimental.

- Overview

- Crea nuevas relaciones, indica que tuplas hacen parte de una relación
- Se denota a partir de Variable o Tupla
- No hay un orden
- No-procedimental y es declarativo.
- Se apoya en la lógica del primer orden, lógica del predicado.
- Es un lenguaje formal.



- Símbología

- Denotamos los consultas de la siguiente forma:

$$\{ t \mid P(t) \} = R$$

"Relación"

"Atomo"

```

    +-----+
    |       |
    |   Variable   "Atomo"
    |       |
    |       +-----+
    |       |   tupla
    |       |
    |       +-----+
    |       |   Dominio
    |       |
    |       +-----+
    |       |   Atomo (es una formula)
    |       |
    |       +-----+
  
```

- Operadores: $\forall, \exists, \Rightarrow, \Leftrightarrow, \neg, \vee, \wedge$

$$\begin{array}{c}
 RCT_{(Tuples)} \xrightarrow{\quad} \text{tupla} \\
 \swarrow \qquad \searrow \\
 RC \qquad \qquad \qquad \text{Variables} \\
 \searrow \qquad \swarrow \\
 RCD_{(Domains)} \xrightarrow{\quad} \text{Dominio}
 \end{array}$$

- Definiciones

- P_1 otra vez es una formula.
- $P_1(t)$ es una formula, siendo 't' una tupla.

- RCT

- Specifying a number of tuple variables.

$t_1: \text{EMPLEADO}(id, nombre, apellido)$

$$R_1 = \{ t \mid t.\text{nombre} = \text{"Pedro"} \}$$

t = resulting tuples

$P(t)$ = Conditions used to fetch 't' (Predicado)

$t[k]$ = tuple variable. Quantified with \exists or by a \forall

↳ It can also be free

- $P(b)$ has the form of

- $t \in R$

- $t \in \text{EMPLEADO}(id, nombre)$

$$R_2 = \{t \mid t \in \text{Empleado}\}$$

- $t[k] \propto u[w]$

- $t \in R_3 = \{t \mid t.id = u.id \wedge u \in \text{Empleado} \wedge t \in \text{Acceso}\}$

- $\text{ACCESO}(id, password)$

- $t[k] \propto C_{\text{constante}}$

- $t \in \text{ESTUDIANTE}(Codigo, nombre, apellido)$

$$R_4 = \{t \mid t \in \text{Estudiante} \wedge t.apellido = 'Ruiz'\}$$

- TCD

- $P(t)$ has the following formats:

- $\langle t_1, t_2, \dots, t_n \rangle \in R \circ (RC(t_1, t_2, \dots, t_n))$

- $t \in \text{Empleado}(id, nombre, apellido)$

$$R_1 = \{t \mid \text{Empleado}(t, "Pedro", "Ruiz")\}$$

- $K \propto W$

- $E \in \text{Estudiante}(Cod, nombre, apellido)$

$$R_4 = \{C, N \mid \text{Estudiante}(C, N, apellido) \wedge$$

- $Acceso(id, A, contraseña) \wedge$

$$\underline{C = A}\}$$

- $\text{Acceso}(id, cod, contraseña)$

• $K \propto C'$

• E_j : Empleado (i , nombre, apellido)

$$T_C = \{ K \mid \text{Empleado}(K, \text{nombre}, \text{apellido}) \}$$

$$A = 'Juan'$$

- Relational Completeness

- Un lenguaje es relativamente completo si es tan expresivo como el álgebra relacional. Si incluye todos sus expresiones.

- Debe tener

- Selección

- Proyección

- Cartesian Product (Join)

- Unión

- Substracción

Theory

Relational Calculus

Relational Algebra

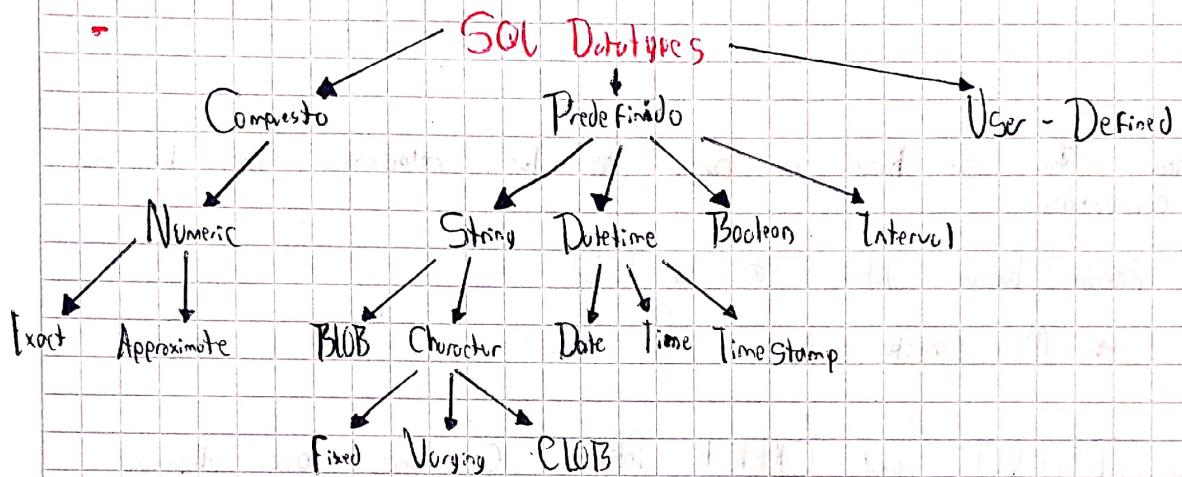
Relational Model

Now

RESTRICCIONES DE INTEGRIDAD

- SQL contains the dual features of DDL and DML, and also a query language (Consulta)

- SQL Datatypes



- Compuestos

- Puedo guardar variables dentro de ellos (Como un atributo compuesto)

- User-Defined

- Dominios creados por el usuario (Ej: Una dirección que tiene una calle y un número).

- Restricciones de integridad

- They provide a way to ensure that changes made by authorized users do not result in a loss of data consistency.

• Restricciones de integridad de dominio

- Set of rules or conditions to restrict the kind of attributes or values a column can hold in the database table.

• Restricciones de integridad de dominio

- "El dominio debe ser igual a la llave foránea de la llave primaria"

-
- Regla lógica que se hace a cargo de las relaciones en que son lógicamente correctas.

- "Uno nunca debería borrar datos de la BD"

- Llave foránea → Me permite Nulos

- NOT NULL, DEFAULT, and CHECK integrity Constraints are allowed.

↑
It prevents null values

↑
Provides a default value

↑
Limits the value range.

- UNIQUE and PRIMARY KEY are allowed as Integrity Constraints.

- NO ACTION → Cuando actualizas un registro, que no se haga nada en los demás en esa misma relación

- CASCADE → Que se propague una acción de modificación o eliminación en un registro a todos sus llaves foráneas.

- SET NULL →

• Secuencias

•

• Disparadores

- Son mecanismos útiles para alertar a los usuarios o para realizar de manera automática ciertas tareas cuando se cumplen determinadas condiciones.

- Sirven como restricciones extra.

- Cuando no usar disparadores

- Resolver problemas que ya tienen otro mecanismo de resolución.
- Para proporcionar datos consolidados, es mejor usar vistas materializadas.
- No redicuar copias con disparadores.

Ricember 6 del 2022

• Transacciones

- Si se hace un cambio en una BD, que esto siga siendo consistente.
- **Correctitud:** Que un registro ingresado quede consistente en la BD.
- "Serie" de pasos que tiene una BD de un estado consistente, en otro estado consistente. Tiene las propiedades ACID (Atomicidad, Consistencia, etc.)

• START ~ ~

~ ~ ~ } Todo esto se realiza aisladamente.

COMMIT TRANSACTION;

"**IF Balance < 0 THEN
ROLLBACK TRANSACTION
END IF**"

- Propiedades ACID

- **Atomicidad:** Una transacción se ejecuta o no se ejecuta.
- Si hay una falla, no se debe de hacer.

- **Consistencia:** Que se cumplan todos los restricciones de integridad.

- **Aislamiento:** Los transacciones concurrentes se ejecutan de forma aislada entre sí.

- **Durabilidad:** Persistencia garantizada de todos los datos.

2. Funciones

• Funciones

- Son bloques de instrucciones SQL que devuelven alguna tarea.

• SELECT

```
    Col1,  
    Col2,  
    function_name()  
FROM TABLE_NAME;
```

- Tienen un nombre, cuerpo de función y un retorno.

• CREATE OR REPLACE FUNCTION *Name*
 RETURNS *integer* AS \$total\$ → Return

{ } Body

\$total\$ LANGUAGE PLPGSQL;

Language.

3. Procedimientos

• CREATE OR REPLACE PROCEDURE *totalRecords()*

PIGEN

{ } Body

END;

- Es opcional retornar algo

- Su fin es ejecutar instrucciones SQL, puede ser almacenado en la BD

SQL Assessment

1. Π_{name_p} (Products)
2. $\Pi_{name_p, price}$ (Products)
3. $\Pi_{name_p} (\exists price \leq 200 \text{ (Products)})$
4. $\exists price < 200 \text{ AND } price > 60 \text{ (Products)}$
- 5.

1. SELECT name_p FROM Products
2. SELECT name_p, price FROM Products
3. SELECT name_p FROM Products WHERE price <= 200
4. SELECT * FROM Products WHERE price Between 60 AND 200
5. SELECT Avg(price) FROM Products
6. SELECT Avg(price) FROM Products WHERE supplier_id = 2
7. SELECT Count(*) FROM Products WHERE price >= 180

8. Obtenga el precio promedio de los productos de cada fabricante, mostrando el nombre del Proveedor

FROM Products, Supplier
VP BY Supplier_id
TIE Supplier_id

SELECT Avg(price), name_s
FROM Products, Supplier
GROUP BY supplier_id

SUBCONSULTAS

- Es una consulta dentro de otra consulta :P.
- **Daridon:** Un valor, Un solo tuplo / Una sola columna, Varias columnas / Una o varias tuplas.
- Puede o no existir relación entre una consulta externa y la interna
- Correlacionado y No Correlacionado. **3 Tipos**
- **Para que los uso?** Para comprobar pertenencia, Si mis tuplos pertenecen al conjunto inicial.
- Comparaciones de conjuntos
- Cálculo de cardinalidades de conjuntos
- Definición de conjuntos de tuplos para una vista.
- **IMPORTANTE:** WHERE, HAVING, IN.

- **Página:** SQLite Online. Com

- **Ej:** **¿Por qué el Constraint? (Cuestión de control)**
- ```
SELECT nombre
 FROM Asignatura
 WHERE creditos = (
 SELECT MAX(creditos) FROM asignatura
);
```
- Us vistes son tablas virtuales derivadas de otras
- **Necesita:** Un nombre, Una lista de atributos y Consulta para especificar.
   
Puedes tener listas de atributos.
- Cuando utilizo una vista, ésta se oculta automáticamente.

## FACTOR DE BLOQUEO

- **Blague:** Registro físico o blague es la continuidad de datos más permanente que se puede transferir en una operación de entrada y salida.

Los archivos se dividen en registros, que se dividen en campos.

- Un orden resibe en memoria Secundaria

- 4 Byte  $\sim$  30  $\sim$  10  $\sim$  2  $\sim$

$$\text{Langford's rule (TCL): } 4 + 30 + 10 + 2 = 46$$

$$\text{Fator de Bloco} (\text{FB}) = \frac{\text{Tamanho do bloco}}{\text{Tamanho da estrutura}} / \text{tamanho do bloco}$$

"Bloco"  $512 / 46 = 11,13$

i 1000 Pedias?

$$1000 / 11 = 90,90 \approx 91$$

$$\text{Especie real: } 91 \times 512 = 46592$$

## - Fonction de bloquage

- Característica importante. [1] Número de registros lógicos que puede contener un registro Físico se denomina **fusión de bloques**.

- Registro lógico > Registro físico } Registros populados

" " = " } FB=2, los registros no están bloqueados

- " < " , 3 FB > 1, 105 registers option bit-words

$$A = B_{RF} = 1$$

$$B(F_{BF}) = 10$$

$$A = \frac{10^6}{a}$$

$$B = \frac{10^6}{10} = 10^5$$

Este es el  
mejor  
Mejor sol  
rendimiento de la  
lectura de Disco duro.

Scribd

- Como se calcula el factor de Blotting (CBF)

- Se necesiten los Signos Símbolos:

$\beta$  = Turnio byte

RC = "registro "

c: Cada uno de registran del archivo, ficheros o tablas

$$\overline{B}_{TGF} = \frac{\overline{B}}{R}$$

$$\frac{1074}{54} \approx 19$$

$$N = \frac{f}{b r F}$$

## Tessardo (1881)

$$\frac{N}{2} \approx 100$$

Bustleida binaria  $\log_2(N)$

# INDICES

-Búsqueda sin índices (Heap)

## Desordenejo

# Orlando

|     | Desordenado |   |   | Ordenado |   |   |
|-----|-------------|---|---|----------|---|---|
|     | 1           | 2 | 3 | 4        | 5 | 6 |
| 9   |             |   |   |          |   |   |
| 5   |             |   |   |          |   |   |
| 13  |             |   |   |          |   |   |
| 9   |             |   |   |          |   |   |
| 8   |             |   |   |          |   |   |
| 6   |             |   |   |          |   |   |
| 1   |             |   |   |          |   |   |
| 2   |             |   |   |          |   |   |
| 3   |             |   |   |          |   |   |
| 4   |             |   |   |          |   |   |
| 5   |             |   |   |          |   |   |
| 6   |             |   |   |          |   |   |
| 7   |             |   |   |          |   |   |
| 8   |             |   |   |          |   |   |
| 9   |             |   |   |          |   |   |
| 10  |             |   |   |          |   |   |
| 11  |             |   |   |          |   |   |
| 12  |             |   |   |          |   |   |
| 13  |             |   |   |          |   |   |
| 14  |             |   |   |          |   |   |
| 15  |             |   |   |          |   |   |
| 16  |             |   |   |          |   |   |
| 17  |             |   |   |          |   |   |
| 18  |             |   |   |          |   |   |
| 19  |             |   |   |          |   |   |
| 20  |             |   |   |          |   |   |
| 21  |             |   |   |          |   |   |
| 22  |             |   |   |          |   |   |
| 23  |             |   |   |          |   |   |
| 24  |             |   |   |          |   |   |
| 25  |             |   |   |          |   |   |
| 26  |             |   |   |          |   |   |
| 27  |             |   |   |          |   |   |
| 28  |             |   |   |          |   |   |
| 29  |             |   |   |          |   |   |
| 30  |             |   |   |          |   |   |
| 31  |             |   |   |          |   |   |
| 32  |             |   |   |          |   |   |
| 33  |             |   |   |          |   |   |
| 34  |             |   |   |          |   |   |
| 35  |             |   |   |          |   |   |
| 36  |             |   |   |          |   |   |
| 37  |             |   |   |          |   |   |
| 38  |             |   |   |          |   |   |
| 39  |             |   |   |          |   |   |
| 40  |             |   |   |          |   |   |
| 41  |             |   |   |          |   |   |
| 42  |             |   |   |          |   |   |
| 43  |             |   |   |          |   |   |
| 44  |             |   |   |          |   |   |
| 45  |             |   |   |          |   |   |
| 46  |             |   |   |          |   |   |
| 47  |             |   |   |          |   |   |
| 48  |             |   |   |          |   |   |
| 49  |             |   |   |          |   |   |
| 50  |             |   |   |          |   |   |
| 51  |             |   |   |          |   |   |
| 52  |             |   |   |          |   |   |
| 53  |             |   |   |          |   |   |
| 54  |             |   |   |          |   |   |
| 55  |             |   |   |          |   |   |
| 56  |             |   |   |          |   |   |
| 57  |             |   |   |          |   |   |
| 58  |             |   |   |          |   |   |
| 59  |             |   |   |          |   |   |
| 60  |             |   |   |          |   |   |
| 61  |             |   |   |          |   |   |
| 62  |             |   |   |          |   |   |
| 63  |             |   |   |          |   |   |
| 64  |             |   |   |          |   |   |
| 65  |             |   |   |          |   |   |
| 66  |             |   |   |          |   |   |
| 67  |             |   |   |          |   |   |
| 68  |             |   |   |          |   |   |
| 69  |             |   |   |          |   |   |
| 70  |             |   |   |          |   |   |
| 71  |             |   |   |          |   |   |
| 72  |             |   |   |          |   |   |
| 73  |             |   |   |          |   |   |
| 74  |             |   |   |          |   |   |
| 75  |             |   |   |          |   |   |
| 76  |             |   |   |          |   |   |
| 77  |             |   |   |          |   |   |
| 78  |             |   |   |          |   |   |
| 79  |             |   |   |          |   |   |
| 80  |             |   |   |          |   |   |
| 81  |             |   |   |          |   |   |
| 82  |             |   |   |          |   |   |
| 83  |             |   |   |          |   |   |
| 84  |             |   |   |          |   |   |
| 85  |             |   |   |          |   |   |
| 86  |             |   |   |          |   |   |
| 87  |             |   |   |          |   |   |
| 88  |             |   |   |          |   |   |
| 89  |             |   |   |          |   |   |
| 90  |             |   |   |          |   |   |
| 91  |             |   |   |          |   |   |
| 92  |             |   |   |          |   |   |
| 93  |             |   |   |          |   |   |
| 94  |             |   |   |          |   |   |
| 95  |             |   |   |          |   |   |
| 96  |             |   |   |          |   |   |
| 97  |             |   |   |          |   |   |
| 98  |             |   |   |          |   |   |
| 99  |             |   |   |          |   |   |
| 100 |             |   |   |          |   |   |

"No hay un atributo que me permita identificar un registro"

Complejidad:  $O(n)$  → Recorro uno a uno para encontrar un registro.

## - Índices

- Un índice es una **estructura de datos**.
- Mejora la velocidad de los operaciones de consulta.
- La clave está en la organización de la información.
- **Clave de ordenación:** Clave que ordena el índice
- Puedo construir un "orden" de mis registros a partir de mi índice.

### a) Índices ordenados.

Basados en una disposición ordenada de los valores. (Alfabético, número, etc.)

### b) Índices Asociativos

Se buscan en una distribución uniforme de los valores a través de una serie de cujos. El valor asignado a cada cujo está determinado por una función de asociación. (hash function)

## - Índices ordenados

### a) Índices primarios:

Es un archivo ordenado secuencialmente y cuya clave de búsqueda es independiente del orden general del archivo. Es común usar la llave primaria.

### b) Índices Secundarios:

Índice cuya clave de búsqueda especifica un orden distinto del orden secuencial del archivo.

### c) Índice disperso:

Solo se crea un registro índice para algunos de los valores clave de la búsqueda.

### d) Índice denso

Aparece un registro índice para cada valor de la tabla.

### e) Índice B+T

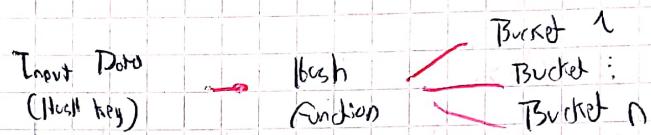
Estructura de datos de forma de árbol balanceado

- ¿Cómo se crea un índice?

```
CREATE INDEX nombre_index
ON nombre_table
USING BTREE
```

- Índices Asociativos

- Distribución Uniforme: Se asigna a cada clave el mismo número de valores de los que se le asignan dentro del conjunto de todos los posibles valores de la misma.



- Distribución Aleatoria: Cada clave tiene el mismo número de valores en un caso promedio.

- ¿Cómo se crea un índice Hash?

USING HASH;

- ¿Elección del tipo de índice?

- Tiempo de acceso
- Tiempo de borrado
- Espacio adicional requerido.

P.D.: En un intervalo es mejor usar una Hash Table. Un BT-tree.

Relax  $\rightarrow$   $P_{\text{intra}}$   
and others

$$\cdot C_{11} = P_S + P_A + P_G - P_Z$$

$$C_{12} = \sim$$

$$C_{21} = \sim$$

$$C_{22} = P_S + P_A - P_Z - P_G$$

