





Bases de de Datos 1



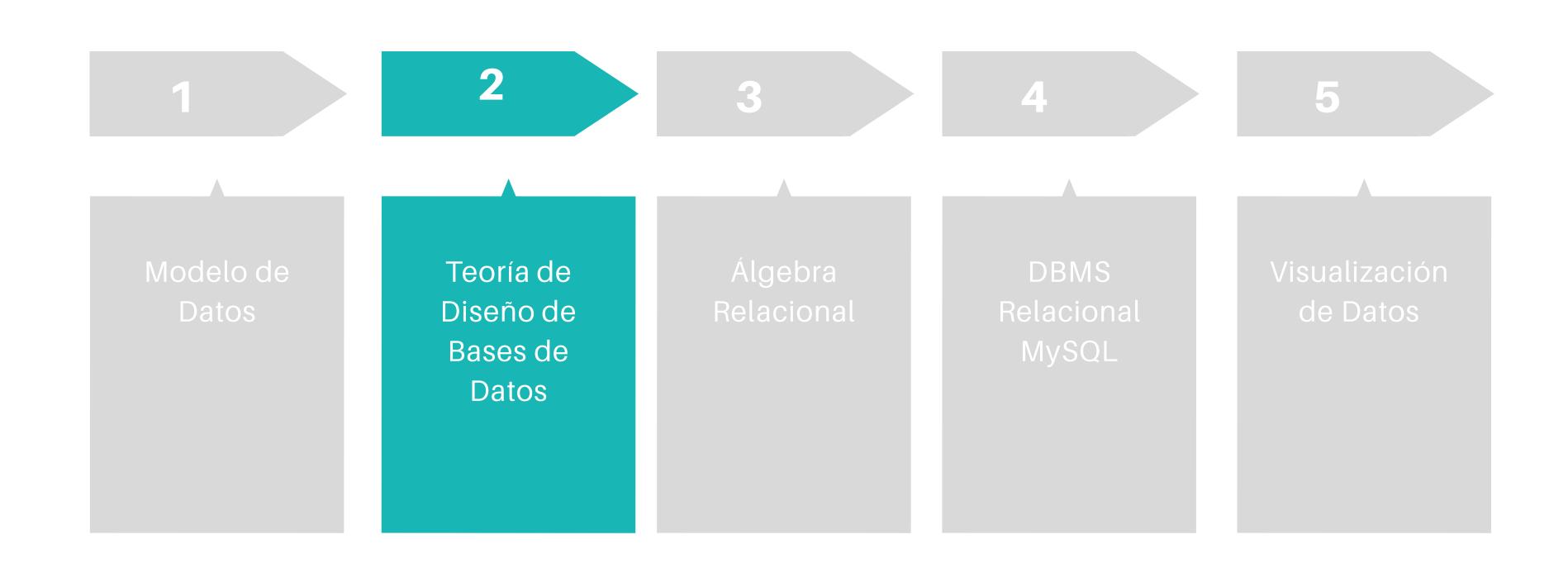
Alejandra Lliteras Prof. Titular



Federico Orlando Prof. Adjunto

TEMAS GENERALES

Bases de Datos 1



Uno de los principales principios en el diseño de bases de datos es la NORMALIZACION

- Organiza los datos siguiendo reglas

 Minimiza redundancia

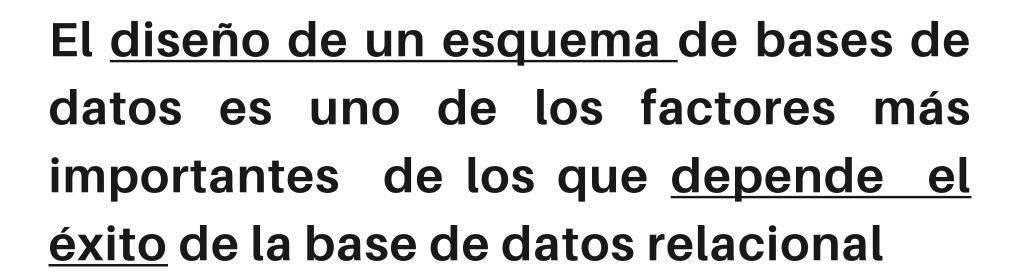
 Reduce anomalías
- Puede mejorar la mantenibilidad y según el caso, la performance



Un mal diseño puede implicar la necesidad de REFACTORIZAR la base de datos

Requiere de mucho tiempo y expertise

Deuda técnica





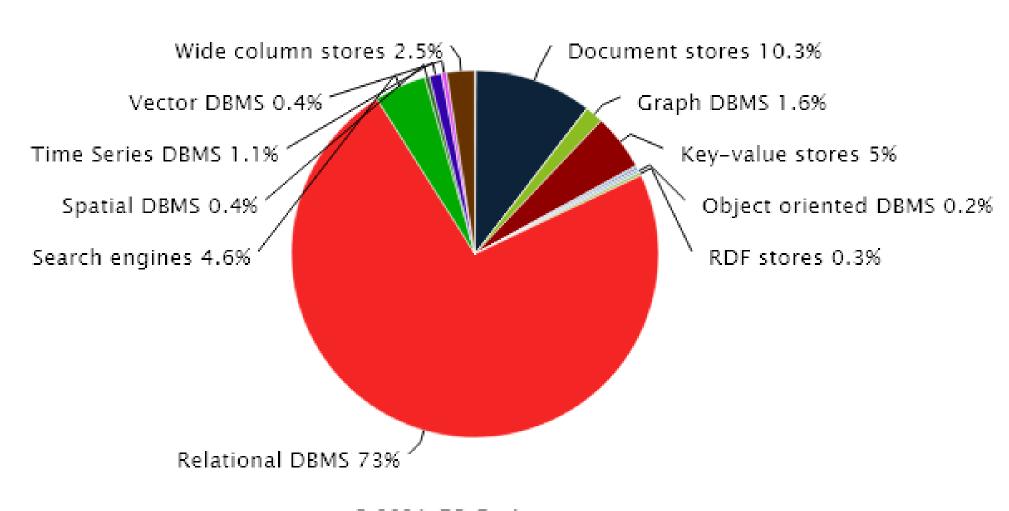
La NORMALIZACION es un paso clave en el diseño de una base de datos

- Toma una <u>relación grande</u> como entrada y la <u>descompone</u> en relaciones más pequeñas las cuales están libres de redundancia de datos y otras anomalías como la de inserción/eliminación
 - La descomposición se realiza siguiendo reglas/pasos
- Puede ser manual o automática



El modelo relacional, sigue siendo el modelo dominante en la industria

Ranking scores per category in percent, August 2024



DB-Engines Ranking of Relational DBMS

Aug 2024	Rank Jul 2024	Aug 2023	DBMS	Database Model
1.	1.	1.	Oracle 🚹	Relational, Multi-model
2.	2.	2.	MySQL 🚹	Relational, Multi-model
3.	3.	3.	Microsoft SQL Server 🚹	Relational, Multi-model
4.	4.	4.	PostgreSQL 🔠	Relational, Multi-model
5.	5.	1 8.	Snowflake 🗄	Relational
6.	6.	4 5.	IBM Db2	Relational, Multi-model
7.	7.	7.	SQLite 🚹	Relational
8.	8.	4 6.	Microsoft Access	Relational
9.	9.	9.	MariaDB 🔠	Relational, Multi-model

SÍNTESIS

- El éxito de una base de datos relacional depende del diseño de su esquema
- Existen procesos manuales y automáticos
- Con impacto en la industria, además de serlo en la academia
- Tema relevante y actual
- El modelo relacional domina el mercado

Teoría de Diseño de Bases de Datos Relacionales

Conceptos generales

TEMAS Y SUBTEMAS

Teoría de Diseño de Bases de Datos Relacionales

Conceptos Generales:

- Anomalía
- Dependencia Funcional
- Dependencia Funcional Trivial
- Clave
 - Clave Candidata
 - Super Clave

Teoría de Diseño de Bases de Datos Relacionales

Anomalía

Normalización:

Toma una relación grande como entrada y la descompone en relaciones más pequeñas las cuales están libres de redundancia de datos y otras <u>anomalías</u> como la de inserción/eliminación

Anomalía

Normalización:

Toma una relación grande como entrada y la descompone en relaciones más pequeñas las cuales están libres de redundancia de datos y otras <u>anomalías</u> como la de inserción/eliminación



Problema que surge a raíz del diseño de una relación

Anomalía

- El código de jefe es único en el sistema.
- Una persona puede trabajar en más de un departamento y en cada uno de ellos posee un código de empleado.
- El código de empleado no se repite ni por departamento, ni entre departamentos.
- FingDepto es la fecha de ingreso a un departamento

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

- El código de jefe es único en el sistema.
- Una persona puede trabajar en más de un departamento y en cada uno de ellos posee un código de empleado.
- El código de empleado no se repite ni por departamento, ni entre departamentos.
- FingDepto es la fecha de ingreso a un departamento

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

- El código de jefe es único en el sistema.
- Una persona puede trabajar en más de un departamento y empleado.
- El código de empleado no se repite ni por departamento, ni enti-
- FingDepto es la fecha de ingreso a un departamento





Anomalía de Redundancia:

Información que se repite innecesariamente en diferentes tuplas

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



Anomalías de actualización:

Se puede actualizar el valor en una tupla, sin actualizar los de otras tuplas

dni	nombre	domicilio	depto	fIngDepto	codEmpDepto	jefe
dni1	Juan	10 Nro 222	Compras	2016-01-11	E1	J1
dni1	Juan	12 Nro 222	Liq. Sueldos	2015-02-01	E2	J2
dni2	Maria	3 Nro. 214	Compras	2014-05-01	E3	J1
dni3	José	3 Nro. 214	Compras	2015-01-08	E4	J1



Anomalías de inserción:

Insertar valores en ciertos atributos de una relación y no en otros me produce valores nulos

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

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



borrar ciertos valores de una tupla, puede llevarme a perder la información de la tupla completa

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

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

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

Anomalía

SINTESIS



- El código de jefe es único en el sistema.
- Una persona puede trabajar en más de un departamento y en cada uno de ellos posee un código de empleado.
- El código de empleado no se repite ni por departamento, ni entre departamentos.
- FingDepto es la fecha de ingreso a un departamento

Teoría de Diseño de Bases de Datos Relacionales

Dependencia Funcional

- Concepto fundamental en Normalización
- Es una RESTRICCIÓN entre subconjuntos de atributos de una relación

Dependencia Funcional



• Si dos tuplas (t1 y t2) de una relación R, coinciden en todos los atributos A1, A2,...,An; entonces DEBEN también coincidir en los atributos B1, B2,..,Bm. Para toda tupla de R.

-Esto se escribe

A1, A2,...,An -> B1, B2,...,Bm

-Y se lee

A1, A2,...,An "determina funcionalmente a" B1, B2,...,Bm



• Si dos tuplas (t1 y t2) de una relación R, coinciden en todos los atributos A1, A2,...,An; entonces DEBEN también coincidir en los atributos B1, B2,..,Bm. Para toda tupla de R.

-Esto se escribe

-Y se lee

A1, A2,...,An "determ

Cuando en R se cumple una df, estamos indicando una restricción sobre toda la relación Ry no sólo sobre algunas tuplas de R.

Dependencia Funcional (df)

Dicho de otra manera:

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

Dependencia Funcional (df)

Ejemplo 1:

–Dada la relación

PERSONA(dni, nombre, edad, fechaNacimiento)

Dependencia Funcional (df)

Ejemplo 1:

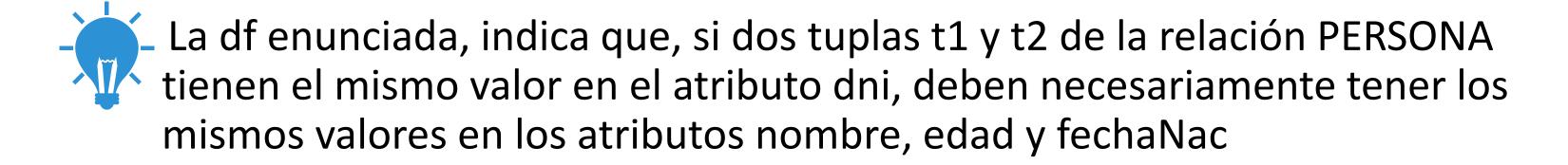
–Dada la relaciónPERSONA(dni, nombre, edad, fechaNacimiento)

–Y valga en PERSONA la df: dni->nombre,edad,fechaNac

Dependencia Funcional (df)



- –Dada la relación
 PERSONA(dni, nombre, edad, fechaNacimiento)
- –Y valga en PERSONA la df: dni->nombre,edad,fechaNac



Dependencia Funcional (df)



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

Dependencia Funcional (df)

```
Ejemplo 2:
```

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

-Se pueden enunciar las siguientes dfs

Dependencia Funcional (df)

- Ejemplo 2:
 - —Dada la relación: PERSONA(dni, nombre, edad, fechaNac, nroLegajo)
 Donde
 - Una persona posee un único número de legajo asignado
 - Un número de legajo pertenece a una sola persona

- -Se pueden enunciar las siguientes dfs
- df1) dni -> nombre, edad, fechaNac, nroLegajo df2) nroLegajo -> nombre, edad, fechaNac, dni

Dependencia Funcional (df)

Ejemplo 3:

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

Dependencia Funcional (df)

Ejemplo 3:

- –Dada la relación: PERSONA(dni, nombre, edad, fechaNac, nroLegajo, carrera)»Donde
 - Una persona puede cursar diversas carreras
 - Nombre indica como se llama la persona
 - Una persona posee un único número de legajo asignado para cada carrera que cursa
 - Un número de legajo pertenece a una sola persona de una carrera
- -Se pueden enunciar las siguientes dfs

Dependencia Funcional (df)



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

```
df1) dni -> nombre, edad, fechaNac
```

df2) nroLegajo, carrera -> dni

df3) dni, carrera -> nroLegajo

Teoría de Diseño de Bases de Datos Relacionales

Dependencia Funcional TRIVIAL

Caso especial para una dependencia funcional

Dependencia Funcional Trivial



Dependencia Funcional trivial

Es una df de la forma:

Tal que:

$$\{B1, B2,...,Bm\} \subseteq \{A1, A2,...,An\}$$

Dependencia Funcional trivial



–Dada la relación:

CONTRATADO(nroContradado, dni, nombrePersona, inicioActividad)

Donde valen las siguientes dependencias funcionales:

- df1) dni -> nombrePersona
- df2) nroContratado, dni -> inicioActividad

Dependencia Funcional trivial



-Dada la relación:

CONTRATADO(nroContradado, dni, nombrePersona, inicioActividad)

Donde valen las siguientes dependencias funcionales:

- df1) dni -> nombrePersona
- df2) nroContratado, dni -> inicioActividad

Algunas dependencias funcionales triviales válidas en CONTRATADO son:

Dependencia Funcional trivial



–Dada la relación:

CONTRATADO(nroContradado, dni, nombrePersona, inicioActividad)

Donde valen las siguientes dependencias funcionales:

- df1) dni -> nombrePersona
- df2) nroContratado, dni -> inicioActividad



Algunas dependencias funcionales triviales válidas en CONTRATADO son:

dft1) dni-> dni

Dependencia Funcional trivial



–Dada la relación:

CONTRATADO(nroContradado, dni, nombrePersona, inicioActividad)

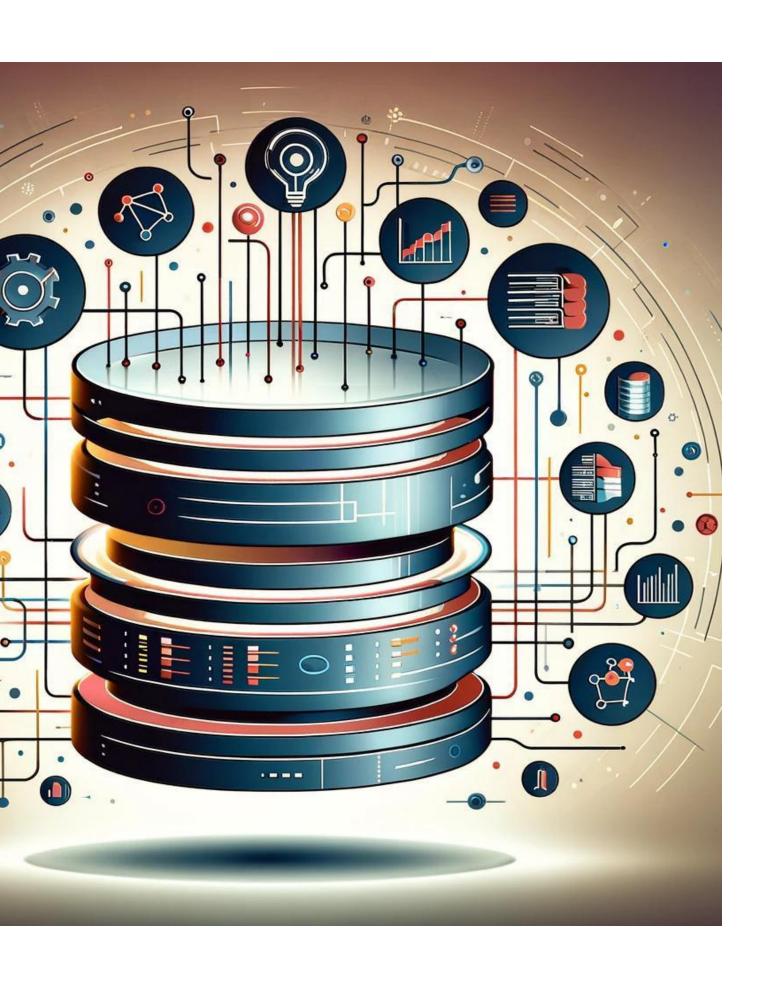
Donde valen las siguientes dependencias funcionales:

- df1) dni -> nombrePersona
- df2) nroContratado, dni -> inicioActividad



Algunas dependencias funcionales triviales válidas en CONTRATADO son:

dft1) dni-> dni dft2) nroContratado, dni -> nroContratado



Bibliografía de la clase

Bibliografía

- Date, C. J. (2019). Database design and relational theory: normal forms and all that jazz. Apress.
- Garcia-Molina, H. (2008). *Database systems: the complete book*. Pearson Education India.
- Ullman, J. D. (1988). Principles of database and knowledge-base systems.
- Albarak, M., Bahsoon, R., Ozkaya, I., & Nord, R. L. (2020). Managing Technical Debt in Database Normalization. IEEE
 Transactions on Software Engineering.
- Jadhav, R., Dhabe, P., Gandewar, S., Mirani, P., & Chugwani, R. (2020). A New Data Structure for Representation of Relational Databases for Application in the Normalization Process. In *Machine Learning and Information Processing* (pp. 305-316). Springer, Singapore.
- Ghawi, R. (2019, May). Interactive Decomposition of Relational Database Schemes Using Recommendations. In International Conference: Beyond Databases, Architectures and Structures (pp. 97-108). Springer, Cham.
- Stefanidis, C., & Koloniari, G. (2016, November). An interactive tool for teaching and learning database normalization. In Proceedings of the 20th Pan-Hellenic Conference on Informatics (pp. 1-4).
- Knowledge Base of Relational and NoSQL Database Management Systems https://db-engines.com/en/ranking_trend
- Akhtar, A. (2023). Popularity Ranking of Database Management Systems. arXiv preprint arXiv:2301.00847.

Importante!



Los slides usados en las clases teóricas de esta materia, no son material de estudio por sí solos en ningun caso.