

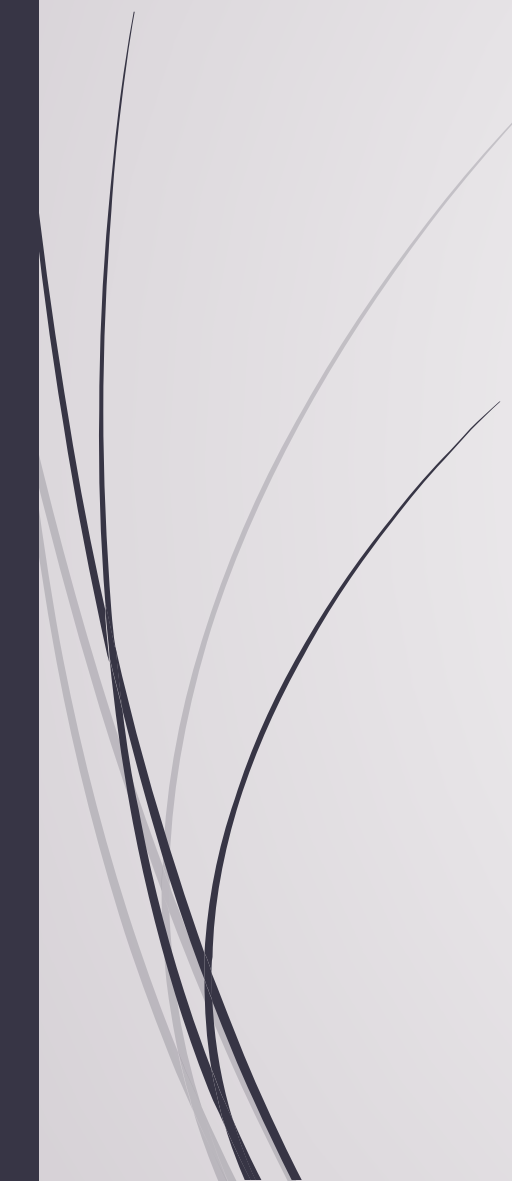
Bases de Datos 1

Clase del 01 de Octubre de 2020

Alejandra Beatriz Lliteras



Contenidos de la materia

- Modelo de datos
 - Conceptos generales
 - Algunos modelos en particular
 - Modelo de Entidades y Relaciones
 - Modelo relacional
 - Transformación entre modelos de datos
 - Álgebra Relacional
 - Operaciones y Consultas
 - Optimización de consultas
 - **Teoría de diseño de bases de datos**
 - **Conceptos generales**
 - Proceso de Normalización
 - SGBD Relacional
 - Conceptos generales de bases de datos
- 

Motivación

Albarak, M., Bahsoon, R., Ozkaya, I., & Nord, R. L. (2020). Managing Technical Debt in Database Normalization. *IEEE Transactions on Software Engineering*.

- ▶ Uno de los principales principios del diseño de bases de datos es la **NORMALIZACION**
- ▶ **Organiza** los datos en relaciones o tablas siguiendo **reglas** específicas para minimizar la redundancia de los datos y en consecuencia, favorecer a la consistencia reduciendo anomalías
- ▶ Un mal diseño puede llevar a la necesidad de **refactorizar** la base de datos y eso suele ser muy caro
- ▶ Los beneficios de la normalización van mas allá de la **calidad** de los datos, pudiendo mejorar la **mantenibilidad** y **performance** de una base de datos
- ▶ En general se ignora por ser un **proceso que requiere tiempo y conocimiento específico**

Motivación

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.

- El **diseño de un esquema** de bases de datos es uno de los factores mas importantes de los que **depende el éxito** de la base de datos relacional
- La normalización es un **paso clave** en el diseño de la base de datos relacional
 - Toma una **relación grande** como entrada y la **descompone** en relaciones mas 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**
 - La automatización de la normalización es de **gran aplicación en la industria** y reduce el costo de la normalización manual



Motivación



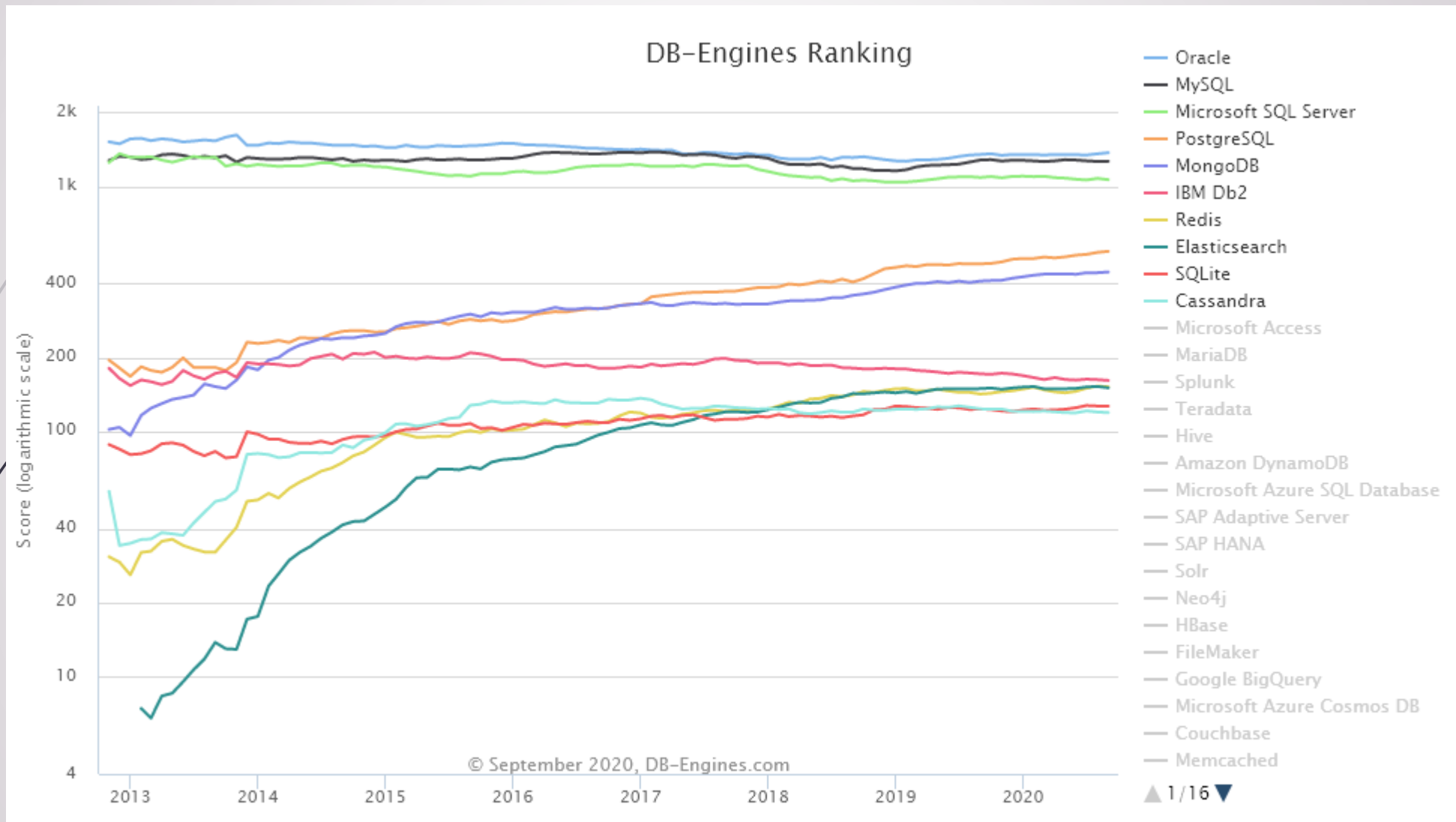
Albarak, M., Bahsoon, R., Ozkaya, I., & Nord, R. L. (2020). Managing Technical Debt in Database Normalization. *IEEE Transactions on Software Engineering*.

- Normalización se aplica a las bases de datos relacionales (no así a otros modelos como por ejemplo NoSQL)
- El modelo relacional, sigue siendo el modelo dominante en la industria
- Utilizado por los 4 principales DBMS
 - Oracle, MySQL, MS SQL Server y PostgreSQL
- También utilizado por DBMS como IBM Db2 y SQLite

Motivación

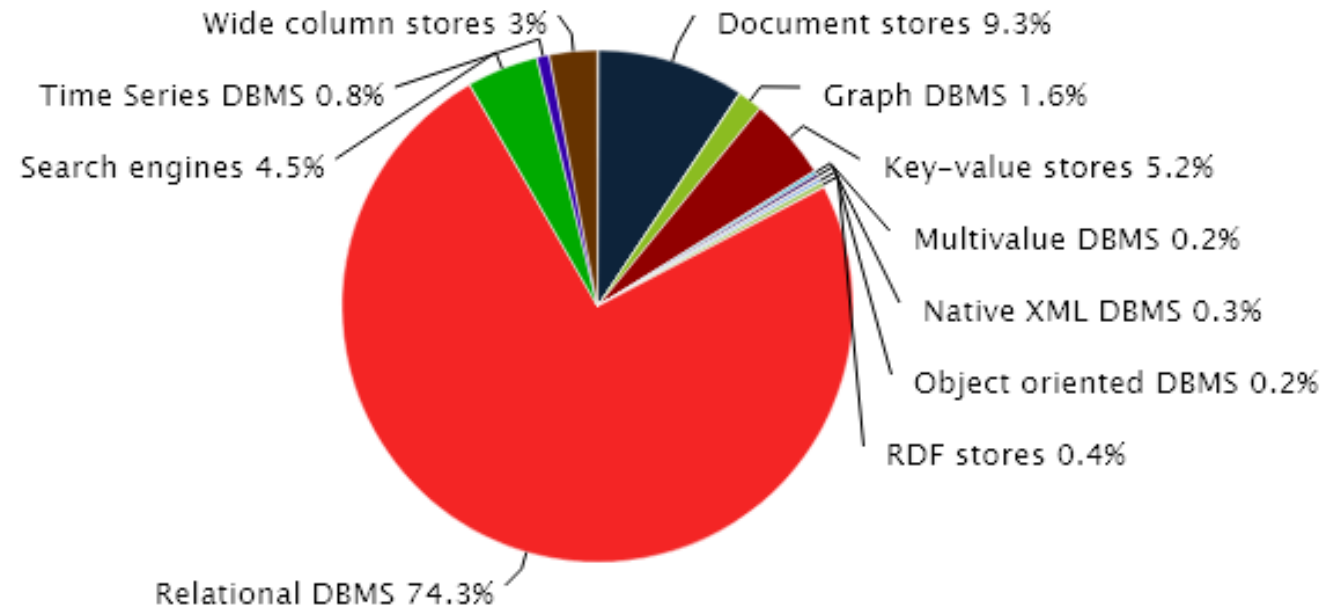
Rank			DBMS	Database Model
Sep 2020	Aug 2020	Sep 2019		
1.	1.	1.	Oracle +	Relational, Multi-model i
2.	2.	2.	MySQL +	Relational, Multi-model i
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model i
4.	4.	4.	PostgreSQL +	Relational, Multi-model i
5.	5.	5.	MongoDB +	Document, Multi-model i
6.	6.	6.	IBM Db2 +	Relational, Multi-model i
7.	7.	↑ 8.	Redis +	Key-value, Multi-model i
8.	8.	↓ 7.	Elasticsearch +	Search engine, Multi-model i
9.	9.	↑ 11.	SQLite +	Relational
10.	↑ 11.	10.	Cassandra +	Wide column

Motivación



Motivación

Ranking scores per category in percent, September 2020



© 2020, DB-Engines.com



Motivación



► En síntesis

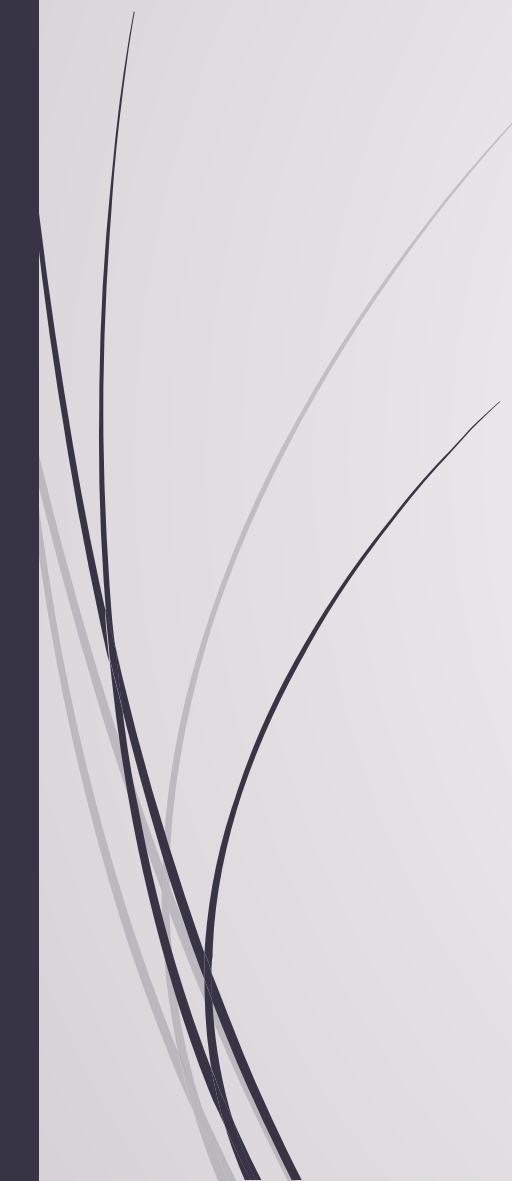
- El éxito de una base de datos relacional depende del diseño de su esquema
- Existen procesos manuales y automáticos
- Con impacto en la industria, además de serlo en la academia
- Tema relevante y actual
- El modelo relacional domina el mercado



Teoría de diseño de Bases de Datos Relacionales



Teoría de diseño de BBDD relacionales

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

Teoría de diseño de BBDD relacionales

El proceso de normalización, toma una **relación grande** como entrada y la **descompone** en relaciones mas pequeñas las cuales están libres de redundancia de datos y **anomalías** como la de inserción/eliminación

► Anomalía

► Problema que surge a raíz del diseño de una relación

Teoría de diseño de BBDD relacionales

PERSONAEMPLEADA(dni, nombre, domicilio, depto, flngDepto, codEmpDepto, jefe)

dni	nombre	domicilio	depto	flngDepto	codEmpDepto	jefe
dni 1	Juan	12 Nro 222	Compras	2016-01-11	E1	J1
dni 1	Juan	12 Nro 222	Liq. Sueldos	2015-02-01	E2	J2
dni 2	Maria	3 Nro. 214	Compras	2014-05-01	E3	J1
dni 3	José	3 Nro. 214	Compras	2015-01-08	E4	J1

- El código de jefe es único en el sistema.
- Una persona puede trabajar en mas 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.

Teoría de diseño de BBDD relacionales

➤ Anomalías

- **Redundancia:** Información que se repite innecesariamente en diferentes tuplas

dni	nombre	domicilio	depto	flngDepto	codEmpDepto	jefe
dni 1	Juan	12 Nro 222	Compras	2016-01-11	E1	J1
dni 1	Juan	12 Nro 222	Liq. Sueldos	2015-02-01	E2	J2
dni 2	Maria	3 Nro. 214	Compras	2014-05-01	E3	J1
dni 3	José	3 Nro. 214	Compras	2015-01-08	E4	J1

Teoría de diseño de BBDD relacionales

► Anomalías

- **Anomalías de actualización:** Se puede actualizar el valor en una tupla, sin actualizar los de otras tuplas

dni	nombre	domicilio	depto	flngDepto	codEmpDepto	jefe
dni 1	Juan	10 Nro 222	Compras	2016-01-11	E1	J1
dni 1	Juan	12 Nro 222	Liq. Sueldos	2015-02-01	E2	J2
dni 2	Maria	3 Nro. 214	Compras	2014-05-01	E3	J1
dni 3	José	3 Nro. 214	Compras	2015-01-08	E4	J1

Teoría de diseño de BBDD relacionales

► Anomalías

- **Anomalías de inserción:** insertar valores en ciertos atributos de una relación y no en otros me produce valores nulos

dni	nombre	domicilio	depto	flngDepto	codEmpDepto	jefe
dni 1	Juan	12 Nro 222	Compras	2016-01-11	E1	J1
dni 1	Juan	12 Nro 222	Liq. Sueldos	2015-02-01	E2	J2
dni 2	Maria	3 Nro. 214	Compras	2014-05-01	E3	J1
dni 3	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?

Teoría de diseño de BBDD relacionales

► Anomalías

- **Anomalías de borrado:** borrar ciertos valores de una tupla, puede llevarme a perder la información de la tupla completa

dni	nombre	domicilio	depto	flngDepto	codEmpDepto	jefe
dni 1	Juan	12 Nro 222	Compras	2016-01-11	E1	J1
dni 1	Juan	12 Nro 222	Liq. Sueldos	2015-02-01	E2	J2
dni 2	Maria	3 Nro. 214	Compras	2014-05-01	E3	J1
dni 3	José	3 Nro. 214	Compras	2015-01-08	E4	J1

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

Teoría de diseño de BBDD relacionales

► Anomalías

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



Teoría de diseño de BBDD relacionales

PERSONAEMPLEADA(dni, nombre, domicilio, depto,
flngDepto, codEmpDepto, jefe)

¿Es un buen diseño?





Teoría de diseño de Bases de Datos Relaciones



DEPENDENCIA FUNCIONAL



Teoría de diseño de BBDD relacionales



Dependencia funcional:

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

Teoría de diseño de BBDD relacionales

► Dependencia Funcional (df)

► Si dos tuplas (t_1 y t_2) de una relación R , coinciden en todos los atributos A_1, A_2, \dots, A_n , entonces DEBEN también coincidir en los atributos B_1, B_2, \dots, B_m . Para toda tupla de R .

► Esto se escribe

$$A_1, A_2, \dots, A_n \rightarrow B_1, B_2, \dots, B_m$$

► Y se lee

A_1, A_2, \dots, A_n "determina funcionalmente a" B_1, B_2, \dots, B_m

Cuando R cumple una df, estamos indicando una restricción sobre toda la relación R y no sobre algunas tuplas de R .

Teoría de diseño de BBDD relacionales

► Dependencia Funcional (df)

► Dicho de otra manera:

► Una dependencia funcional de la forma $X \rightarrow Y$ 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 $t1[y]=t2[y]$.

Teoría de diseño de BBDD relacionales

► Dependencia Funcional (df)

► Ejemplos:

► Dada la relación

► PERSONA(dni, nombre, edad, fechaNacimiento)

► Y valga en PERSONA la

► df: dni->nombre,edad,fechaNac

► La df enunciada, indica que si dos tuplas t1 y t2 de la relación PERSONA tienen el mismo valor en el atributo dni, deben necesariamente tener los mismos valores en los atributos nombre, edad y fechaNac

Teoría de diseño de BBDD relacionales

► Dependencia Funcional (df)

► Ejemplos:

► 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

Teoría de diseño de BBDD relacionales

► Dependencia Funcional (df)

■ Ejemplos:

► 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 para bases de datos relaciones

DEPENDENCIA FUNCIONAL TRIVIAL

Teoría de diseño de BBDD relacionales

► Dependencia Funcional trivial

Es una df de la forma:

$$A_1, A_2, \dots, A_n \rightarrow B_1, B_2, \dots, B_m$$

Tal que:

$$\{B_1, B_2, \dots, B_m\} \subseteq \{A_1, A_2, \dots, A_n\}$$

Teoría de diseño de BBDD relacionales

► Dependencia Funcional trivial

■ Ejemplos:

► Dada la relación:

► CONTRATADO(nroContratado, dni, nombrePersona, inicioActividad)

► Dfs:

► df1) dni -> nombrePersona

► df2) nroContratado, dni -> inicioActividad

Ejemplos de dependencias funcionales triviales:

dni-> dni

nroContratado, dni -> nroContratado



Teoría de diseño para bases de datos relaciones

CLAVE DE UNA RELACIÓN

Teoría de diseño de BBDD relacionales

► Clave de una relación

- Los atributos $\{A_1, A_2, \dots, A_n\}$ son la clave de una relación R si cumplen:
 - $\{A_1, A_2, \dots, A_n\}$ determinan funcionalmente a todos los restantes atributos de la relación R
 - No existe un subconjunto de $\{A_1, A_2, \dots, A_n\}$ que determine funcionalmente a todos los atributos de R –*Esto implica que una clave es un conjunto minimal-*

Teoría de diseño de BBDD relacionales

➤ Clave de una relación

➤ Ejemplo

➤ PERSONA(dni, nombre, edad, fechaNac)

➤ df1: dni->nombre,edad,fechaNac

➤ Clave: {dni}



Teoría de diseño para bases de datos relaciones

CLAVE CANDIDATA DE UNA RELACIÓN

Teoría de diseño de BBDD relacionales

► Clave de una relación /Clave candidata

- En caso de existir dos o mas conjuntos de atributos $\{A_1, A_2, \dots, A_n\}$, $\{B_1, B_2, \dots, B_k\}$, ... $\{N_1, N_2, \dots, N_m\}$ en una relación R tales que
 - $\{A_1, A_2, \dots, A_n\}$ determinan funcionalmente a todos los restantes atributos de la relación R
 - $\{B_1, B_2, \dots, B_k\}$, ... y $\{N_1, N_2, \dots, N_m\}$ también por si mismos determinan al resto de los atributos de R
 - No existe un subconjunto de $\{A_1, A_2, \dots, A_n\}$ o $\{B_1, B_2, \dots, B_k\}$, ... o $\{N_1, N_2, \dots, N_m\}$ que determine funcionalmente a todos los atributos de R

Entonces $\{A_1, A_2, \dots, A_n\}$, $\{B_1, B_2, \dots, B_k\}$, ... $\{N_1, N_2, \dots, N_m\}$ son **CLAVES CANDIDATAS** para la relación R

Teoría de diseño de BBDD relacionales

➤ Clave de una relación /Clave candidata

➤ Ejemplo:

➤ Dada la relación: PERSONA(dni, nombre, edad, fechaNacimiento, 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

Clave candidata 1 (cc1): {dni}

Clave candidata 2 (cc2): {nroLegajo}

Teoría de diseño de BBDD relacionales

► Clave de una relación /Clave candidata

► Ejemplo:

► Dada la relación: PERSONA(dni, nombre, edad, fechaNacimiento, 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

df1) dni -> nombre, edad, fechaNac

df2) nroLegajo, carrera -> dni

df3) dni, carrera -> nroLegajo

Clave candidata 1 (cc1): {nroLegajo, carrera }

Clave candidata 2 (cc2): {dni, carrera }



Teoría de diseño para bases de datos relaciones

SUPERCLAVE DE UNA RELACIÓN

Teoría de diseño de BBDD relacionales

► Superclave de una relación

- “Super conjunto” de una clave
- Los atributos $\{A_1, A_2, \dots, A_n\}$ son la superclave de una relación R si cumplen:
 - $\{A_1, A_2, \dots, A_n\}$ determinan funcionalmente a todos los restantes atributos de la relación R
- Notar que:
 - Una clave esta contenida en una superclave
 - Una superclave no necesariamente es minimal (como lo es la clave por la segunda condición de su definición)

Teoría de diseño de BBDD relacionales

► Superclave de una relación

► Ejemplo

► PERSONA(dni, nombre, edad, fechaNacimiento)

► df1: dni->nombre,edad,fechaNac

► **superclave: {dni, nombre}**



Actividades para el encuentro participativo

Teoría de diseño de BBDD relacionales

Ejercicio A

Dada la relación

VENTAS(codCliente, nombre, codVenta, monto)

► Donde:

- Un cliente realiza muchas compras
- Una compra es realizada por un solo cliente
- El monto representa el valor total de la compra realizada por un cliente
- Determinar las dependencias funcionales (dfs) y clave o clave candidatas válidas en VENTAS

Teoría de diseño de BBDD relacionales

Ejercicio B

Dada la relación

VENTAS1(codCliente, nombreCliente, codVenta, monto)

Donde:

- Un cliente realiza muchas compras
- Una compra puede ser realizada por mas de un cliente
- El monto representa el valor total de la compra realizada por cada cliente
- Determinar las dependencias funcionales (dfs) y clave o clave candidatas válidas en VENTAS1

Teoría de diseño de BBDD relacionales

Ejercicio C

Dada la relación

PERSONAEMPLEADA(dni, nombre, domicilio, depto, flngDepto, codEmpDepto, jefe)

Donde

- *En cada departamento hay un jefe para todos los empleados. Un mismo jefe puede estar asignado a mas de un departamento*
- *Una persona puede trabajar en mas de un departamento y en cada uno de ellos puede tener un código de empleado diferente*
- *El código de empleado no se repite para un mismo departamento, puede repetirse en diferentes departamentos*
- *Domicilio indica el lugar en el que vive una persona. Mas de una persona pueden vivir en el mismo domicilio*
- *Nombre indica la forma en la que se llama una persona. Notar que diferentes personas pueden llamarse igual*

Determinar las dependencias funcionales y clave o clave candidatas válidas en **PERSONAEMPLEADA**

Teoría de diseño de BBDD relacionales

Ejercicio D

Dada la relación

PERSONAEMPLEADA1(dni, nombre, domicilio, depto, flngDepto, codEmpDepto, jefe)

Donde

- Cada persona en un departamento tiene asignado a un jefe. El mismo jefe puede estar asignado a diferentes personas de un departamento o de diversos departamentos
- Una persona puede trabajar en mas de un departamento y en cada uno de ellos puede tener un código de empleado diferente
- El código de empleado no se repite para un mismo departamento, puede repetirse en diferentes departamentos
- Domicilio indica el lugar en el que vive una persona. Mas de una persona pueden vivir en el mismo domicilio
- Nombre indica la forma en la que se llama una persona. Notar que diferentes personas pueden llamarse igual

Determinar las dependencias funcionales y clave o clave candidatas válidas en **PERSONAEMPLEADA1**

Bibliografía de los temas abordados en esta clase

- 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

Important
Message

IMPORTANTE: los slides usados en las clases teóricas de esta materia, no son material de estudio por sí solos.