

BASES DE DATOS

The background of the slide features several overlapping, wavy lines in orange, light blue, and lime green, creating a modern, abstract design.

CLASE 4

Normalización

- La **normalización** fue creada por Codd como una **técnica de diseño** de base de datos en el año 1972
 - Se parte de los atributos y éstos se van agrupando en tablas, según su afinidad
- Es posible aplicarla sobre el esquema, como una **técnica de verificación**
 - Eliminación de las dependencias entre atributos no deseadas

Normalización

- **Definición:** proceso de análisis de los esquemas de tablas dados basado en sus dependencias funcionales y claves primarias
- **Formas normales**
 - Los esquemas de tablas que no cumplan con determinadas condiciones (pruebas normales) se descomponen en esquemas más pequeños que satisfagan dichas pruebas

Normalización

- **Objetivos**
 - Controlar la redundancia de información
 - Evitar pérdidas de información
 - Mantener la consistencia de los datos

Normalización

- La normalización es un proceso deseado, pero no es estrictamente necesario
 - Puede afectar el **tiempo de acceso** a los datos en las tablas en donde se producen cambios
 - Se puede optar por dejar de lado la normalización y **priorizar la performance** final de la BD

Normalización

- Problemas de una mala normalización o una normalización inexistente
 - Anomalías de inserción
 - Anomalías de modificación
 - Anomalías de eliminación

Normalización

- **EMPLEADOS** = (idEmpleado, dni, nombre, salario, fechaIngreso, idDepto, nombreDepto)
- Anomalías de **inserción**
 - Agregar más de un empleado para el mismo depto
 - Repetir el depto
 - Error de nombres
 - Agregar un nuevo depto
 - Datos nulos

Normalización

- **EMPLEADOS** = (idEmpleado, dni, nombre, salario, fechaIngreso, idDepto, nombreDepto)
 - Anomalías de **modificación**
 - Modificar el nombre de un depto
 - Modificar cada uno
 - Error de nombres
 - Anomalías de **eliminación**
 - Eliminar un empleado
 - Puede perderse un depto

Dependencias Funcionales

- Una **dependencia funcional (DF)** representa una **restricción entre atributos** de una tabla de la base de datos
- Dados X , Y atributos, se dice que Y **depende funcionalmente** de X cuando para un valor dado de X siempre se encuentra el mismo valor para el atributo Y
- X e Y pueden representar también un conjunto de atributos

Dependencias Funcionales

- Notación: $X \rightarrow Y$
 - El valor del atributo Y depende funcionalmente del valor del atributo X
 - El valor del atributo X determina el valor del atributo Y
 - Y depende de X
 - X determina a Y
 - Y esta determinado por X
- Una **DF siempre existe** entre los atributos no clave de la tabla respecto del clave

Dependencias Funcionales

- Ejemplo 1:

DOCENTE = (DNI, nombre, dirección, tel, idTitulo)

- Dada la anterior tabla, se establecen las siguientes **DF**:
 - $\text{DNI} \rightarrow \text{nombre}$
 - $\text{DNI} \rightarrow \text{dirección}$
 - $\text{DNI} \rightarrow \text{tel}$
 - $\text{DNI} \rightarrow \text{idTitulo}$

Dependencias Funcionales

- Ejemplo 2:

ALUMNO = (DNI, nombre, dirección, tel, nroAlumno)

- Dada la anterior tabla, se establecen las siguientes **DF**:
 - $\text{DNI} \rightarrow \text{nombre}$
 - $\text{DNI} \rightarrow \text{dirección}$
 - $\text{DNI} \rightarrow \text{tel}$
 - $\text{DNI} \rightarrow \text{nroAlumno}$
 - $\text{nroAlumno} \rightarrow \text{nombre}$
 - $\text{nroAlumno} \rightarrow \text{dirección}$
 - $\text{nroAlumno} \rightarrow \text{tel}$
 - $\text{nroAlumno} \rightarrow \text{DNI}$

Dependencias Funcionales

- **DF** Parcial

- Una **DF** $X \rightarrow Y$ es **parcial** cuando además existe otra **DF** $Z \rightarrow Y$, siendo Z subconjunto de X
 - Un atributo **B** (que no es clave) depende de un subconjunto de **A** (clave)
- Problemas:
 - Repetición de información
 - Puede generar anomalías de inserción, borrado o modificación

Dependencias Funcionales

- **DF** Parcial

- Ejemplo:

PEDIDOS = (idPedido, idProducto, descProducto,
fechaPedido, cantidad)

- ¿Cuáles son las dependencias funcionales?

Dependencias Funcionales

- **DF** Transitiva

- Una **DF** $X \rightarrow Y$ es **transitiva** cuando existe un atributo **Z** tal que $X \rightarrow Z$ y $Z \rightarrow Y$
 - Un atributo **B** (que no es clave) depende de un atributo **C** que tampoco es clave
- Problemas:
 - Repetición de información
 - Puede generar anomalías de inserción, borrado o modificación

Dependencias Funcionales

- **DF** Transitiva

- Ejemplo:

ALUMNO = (idAlumno, nombre, dirección, idCarrera,
nombreCarrera)

- ¿Cuáles son las dependencias funcionales?

Dependencias Funcionales

- **DF** de Boyce-Codd
 - Una **DF** $X \rightarrow Y$ es de Boyce-Codd (DFBC) cuando X no es una **CP** o **CC**, e Y es una **CP** o **CC** (o parte de ella)
 - Un atributo **B** (clave o parte de clave) depende de un atributo **C** que no es clave
 - Problemas:
 - Repetición de información
 - Puede generar anomalías de inserción, borrado o modificación

Dependencias Funcionales

- **DF** de Boyce-Codd

- Ejemplo: en la siguiente tabla se registran las materias que cursan los alumnos y el docente a cargo de cada una de ellas. Cada docente dicta solamente una materia, y una materia es dictada por un único docente

MATERIA = (nombreAlumno, nombreMateria,
nombreDocente)

- ¿Cuáles son las dependencias funcionales?

Normalización

- **Dependencias Multivaluadas**
 - Una **dependencia multivaluada (DM)** entre dos conjuntos de atributos **X** e **Y** de una tabla cualquiera, indica que:
 - Para un valor determinado de X es posible determinar múltiples valores para el atributo Y
 - El conjunto de valores de Y no está relacionado con los valores de X
 - Una **DM** es denotada como **$X \twoheadrightarrow Y$**

Normalización

- **Dependencias Multivaluadas**
 - Una **DM** no representa, en sí, una situación anómala para una BD
 - Sin embargo, establece que los atributos multideterminados en la **DM** son independientes
 - Esta independencia trae como consecuencia un tipo de **redundancia** que no se puede eliminar con las dependencias funcionales

Normalización

- **Dependencias Multivaluadas**
 - Las únicas dependencias multivaluadas que no generan una gran cantidad de repetición de datos son las **triviales**
 - Una **DM** $X \twoheadrightarrow Y$ se considera **trivial** si:
 - Y está multideterminado por el atributo X
 - Y no está multideterminado por un subconjunto de X

Normalización

- **Dependencias Multivaluadas**

- Ejemplo:

DATOS_SUC = (nombreSucursal, propietarioSucursal, empleadoSucursal)

nombreSucursal	propietarioSucursal	empleadoSucursal
La Plata	Gomez	Bertagno
La Plata	Perez	Bertagno
La Plata	Gomez	Cappa
La Plata	Perez	Cappa

- Un **propietario** puede ser dueño de varias sucursales
- Un **empleado** puede trabajar en varias sucursales

Normalización

- **Dependencias Multivaluadas**

NombreSucursal	PropietarioSucursal	EmpleadoSucursal
La Plata	Gomez	Bertagno
La Plata	Perez	Bertagno
La Plata	Gomez	Cappa
La Plata	Perez	Cappa
La Plata	Gomez	Arias
La Plata	Perez	Arias

- Si se agrega un **nuevo empleado** a una sucursal, en **DATOS_SUC** se debe ingresar **dos tuplas**, ya que la sucursal tiene dos propietarios
- Análogamente, si se agrega un nuevo propietario, se debe ingresar una **nueva tupla por cada empleado** de dicha sucursal

Normalización

- **Dependencias Multivaluadas**

- **DATOS_SUC** = (nombreSucursal, propietarioSucursal, empleadoSucursal)
- Tanto el empleado como el propietario dependen directamente de la sucursal:
nombreSucursal \twoheadrightarrow empleadoSucursal
nombreSucursal \twoheadrightarrow propietarioSucursal
- Pero existen otras DM no triviales que generan las repeticiones de datos:
(nombreSucursal, propietarioSucursal) \twoheadrightarrow empleadoSucursal
(nombreSucursal, empleadoSucursal) \twoheadrightarrow propietarioSucursal

Normalización

- **Dependencias Multivaluadas**
 - **DATOS_SUC** = (nombreSucursal, propietarioSucursal, empleadoSucursal)
 - Se descompone el esquema utilizando las **DM triviales** encontradas:
EMP_SUC = (nombreSucursal, empleadoSucursal)
PROP_SUC = (nombreSucursal, propietarioSucursal)
 - Ahora sólo existen **DM triviales** en los esquemas resultantes

Normalización

- **Formas normales**
 - **Primera Forma Normal (1FN)**: todos los atributos tiene cardinalidad 0 o 1
 - **Segunda Forma Normal (2FN)**: una tabla está en 2FN si está en 1FN y no existen dependencias parciales
 - **Tercer Forma Normal (3FN)**: una tabla está en 3FN si está en 2FN y no existen dependencias transitivas

Normalización

- **Formas normales**
 - **Forma Normal de Boyce Codd (FNBC)**: una tabla está en FNBC si está en 3FN y no existen **dependencias de Boyce-Codd**
 - **Cuarta Forma Normal (4FN)**: una tabla está en 4FN si está en FNBC y sólo existen **dependencias multivaluadas triviales**

Ejercicio

Centro de entrenamiento de fútbol

De cada jugador que se entrena en el centro se conoce sus datos personales (apellido, nombre, DNI, domicilio, teléfonos, fecha y lugar de nacimiento) y sus datos futbolísticos (puesto, pie dominante, club actual <si es que tiene>, historial deportivo). De los clubes se conoce su nombre, un email y un teléfono de contacto. El historial deportivo de un jugador debe indicar el club y categoría de cada equipo en el que participó, el período en el que lo hizo, la cantidad de partidos jugados y la cantidad de goles convertidos. Además, a cada jugador se le realiza un examen físico al ingresar al centro, el cual registra su altura, peso y masa muscular. Este examen se repite periódicamente.

El centro lleva también el registro de la performance de los jugadores en una serie de sistemas de entrenamiento (pases, definición, cabezazo, etc). Cada sistema tiene un nombre que lo identifica y una descripción. Periódicamente se registra el puntaje logrado por los jugadores en estos sistemas para seguir su evolución.

Por otro lado, se almacena información de los entrenadores. Se guarda sus datos personales y título <si es que tiene>. Se supone que un entrenador del centro no trabaja simultáneamente en otro club, pero se debe guardar sus anteriores trabajos en clubes (detallando el club, categoría y período).