



# **BASE DE DATOS**

Semana 04

**UPN.EDU.PE**

**SEMANA 04**



# NORMALIZACIÓN DE BASE DE DATOS

# PRESENTACIÓN DE LA SESIÓN

## Logro de la Sesión



Al término de la sesión los estudiantes podrán identificar los conceptos necesarios para crear un modelo de datos entidad-relación extendido y determinar la lógica necesaria para aplicar Normalización, de manera clara y precisa de acuerdo a los esquemas y procedimientos presentados en clase. De la misma forma, resolverán problemas de diagramas entidad-relación extendido y de normalización.



# Recuperando saberes previos



La llave primaria es ...

- a. La llave que permite acceder a la tabla
- b. Cualquier dato numérico
- c. El dato que identifica de forma única cada registro
- d. Una columna de la tabla con valores únicos y que tener valores nulos.

# Recuperando saberes previos



La llave foránea es ...

- a. La llave que identifica varios registros
- b. La que hace referencia a la llave primaria de otra tabla
- c. Es la llave que puede remplazar a la llave primaria
- d. Es la llave que contiene un valor acumulado

# Recuperando saberes previos



¿Qué se entiende por dependencia funcional?

- a. Relación entre tablas
- b. Relación entre atributos
- c. Relación entre bases de datos
- d. Relación entre registros



# **Anomalías en las operaciones de datos**

# Anomalías en las operaciones de datos



## Anomalías en las operaciones de datos

- Efecto secundario inesperado de una operación de fila
  - consecuencia no intencional (a veces bueno, a veces malo)
- Anomalía de modificación:
  - No se pueden modificar sólo los datos deseados
  - Debe modificar más de los datos deseados
- Causa:
  - Redundancia: hechos almacenados varias veces
  - Solución: Eliminar las redundancias (no deseados) para eliminar anomalías



# Tabla de base de datos de una universidad



## Tabla de base de datos de una universidad

StdNo	StdClas	ProgNo	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Toda la base de datos en una tabla (error de principiante)
- ¿encuentra algún error de inserción, eliminación o actualización en la tabla?

# TABLA DE BASE DE DATOS DE UNA UNIVERSIDAD



<u>StdNo</u>	StdClass	<u>ProgNo</u>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- ¿Qué pasa si queremos insertar un curso?
- No se puede insertar un curso nuevo sin un estudiante sin inscribirlo en una programado (ProgNo parte de PK)
- La creación del curso debe ser independiente

# Tabla de base de datos de una universidad



StdNo	StdClass	ProgNo	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- ¿Qué pasa si queremos actualizar la descripción de un curso?
- Cambiar la descripción del curso; debemos cambiar la descripción en cada inscripción del curso
- La actualización se debe realizar en un solo lugar

# TABLA DE BASE DE DATOS DE UNA UNIVERSIDAD



<b>StdNo</b>	<b>StdClas</b>	<b>ProgNo</b>	<b>ProgAnio</b>	<b>ProgNota</b>	<b>CursoNo</b>	<b>CursoDesc</b>
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- ¿Qué pasa si queremos eliminar la inscripción del estudiante S1 en el curso C1 que esta en la primera fila?
- Al eliminar la primera fila; perderemos información sobre el curso C1 y la programación O1
- Debe ser independiente la operacion

# VENTAJAS Y DESVENTAJAS EN LAS ANOMALÍAS DE DATOS



- Más fácil de consultar: sin uniones (joins)
- Más difícil de cambiar: actualizaciones, eliminaciones, insert con problemas que acabamos de analizar.



# **Dependencia Funcional**

# DEPENDENCIA FUNCIONAL



- La teoría de la normalización se basa en el concepto de dependencias, hasta el punto que actualmente se conoce también como teoría de las dependencias.
- La existencia de una dependencia no se puede demostrar, pero si afirmar por observación del mundo real que se trata de modelar.
- Las dependencias nos muestra algunas importantes interrelaciones existentes entre los atributos del mundo real, cuya semántica tratamos de incorporar a nuestra base de datos; son, por lo tanto, invariantes en el tiempo, siempre que no cambie el mundo real del cual proceden.
- En el proceso de normalización será fundamental empezar identificando todas la dependencias funcionales del universo del caso cuyo diseño estamos realizando, y preocuparemos conservar las dependencias funcionales a los largo del proceso, de modo que el esquema relacional resultante de la normalización tenga las mismas dependencias funcionales que el esquema de partida, o un conjunto equivalente.

# Dependencia Funcional



- En una relación **R**, el atributo **B** depende funcionalmente del atributo **A** si cada valor de **B** en **R** está asociado exactamente con un valor de **A**.
- Al atributo **A** se le llama determinante.
- Todos los atributos de una tabla deben depender funcionalmente de la clave. Sin embargo, la dependencia funcional no necesita que un atributo sea una clave para determinar funcionalmente a otros atributos.
- La dependencia funcional también se puede definir de la siguiente manera:

Dada una relación **R**, el atributo **B** depende funcionalmente de **A** únicamente cuando dos tuplas **R** coinciden en su valor **A**, deben coincidir en su valor **B**.





# Dependencia Funcional



**Tabla: Empleado**

Code	Name	City
E1	Claudia	Chiclayo
E2	Sandra	Trujillo
E3	Henry	Cuzco

Code → Name

Code → City

Code → Name, City

- Dado un valor particular de **Code**, hay exactamente un valor correspondiente para **Name**.
- Por ejemplo, para Code **E1** hay exactamente un valor de Name, **Claudia**. Por eso **Name** es funcionalmente dependiente de **Code**.
- De manera similar, existe exactamente un valor de City para cada valor de Code. Por eso el atributo **City** es funcionalmente dependiente de **Code**.
- El atributo **Code** es determinante.
- También se puede decir que Code determina City y Name.

# Dependencia Funcional



**Tabla: Notas**

ID	City	C_Code	Nota
AD0036	London	C1	15
AD0078	New York	C1	16
CC0075	New York	C2	18
CC0097	Florida	C1	13
AD0036	London	C2	15
CC0075	New York	C1	14

Tabla de notas de alumnos del programa de educación a distancia.

- El ID determina funcionalmente la CITY.
- Pero el ID no es candidato a clave. El candidato a clave en este caso, será la combinación de ID y C\_CODE.
- Los atributos ID y C\_CODE son claves foráneas que hacen referencia a tablas que almacenan información del alumno y el curso respectivamente.
- Por lo tanto, aunque ID no es candidato a clave, determina funcionalmente otro atributo (CITY).
- Note que para un valor particular de ID, el valor de CITY es el mismo en cualquier fila.
- Que ID determina funcionalmente la CITY significa que cada alumno esta localizado en una ciudad.
- Esta dependencia funcional (ID determina CITY) también significa que muchos estudiantes están localizados en una ciudad, pero un estudiante esta localizado solo en una ciudad.



## Dependencia Funcional Completa

Se dice que B tiene dependencia funcional completa o plena de A, si depende funcionalmente de A, pero no depende de ningún subconjunto de A.

$(\text{IdPedido}, \text{IdArticulo}) \longrightarrow \text{Precio}$





## Descriptores Equivalentes

- Se dice que dos descriptores X e Y son equivalentes, si se cumple que:

$$X \rightarrow Y$$

$$Y \rightarrow X$$

- Lo cual se puede representar por:

$$X \leftrightarrow Y$$

- Por Ejemplo:

$$\text{IdArticulo} \leftrightarrow \text{NombArticulo}$$

# Dependencia Funcional



## Dependencia Funcional Transitiva

Se dice que C tiene dependencia transitiva respecto de A, a través de B si:

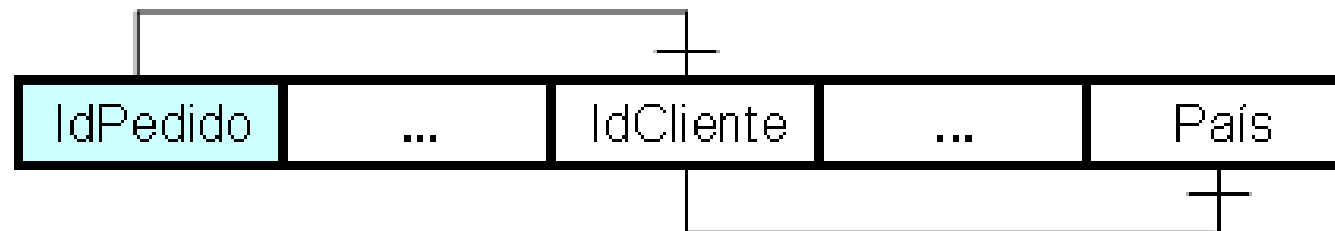
$$A \rightarrow B$$

$$B \rightarrow C$$

Por ejemplo:

$$\text{IdPedido} \rightarrow \text{IdCliente}$$

$$\text{IdCliente} \rightarrow \text{País}$$





# Actividad 1

Para los siguientes casos, por separado, determine su diagrama relacional y sus dependencias funcionales.

1. En la empresa **ABC** se necesita un documento para que los empleados puedan hacer las solicitudes de sus requerimientos de útiles de escritorio para el desarrollo de sus labores.
2. El gerente de **Cines Perú** necesita un documento para registrar la cantidad de personas que han ingresado en cada función de su salas de cine, cuenta con 10 salas, y en una sala se proyectan un promedio de 6 películas por día, depende del tiempo de duración de cada película.

# Dependencias funcionales



- Afirmaciones o Restricciones sobre los datos (reglas de negocio)
- Una dependencia funcional es una restricción de datos en una tabla
  - Clave Primaria, las claves foráneas
  - Una restricción implica una comparación de columnas
- Las restricciones deben plasmar las limitaciones que existen en el modelo de negocio
- Es necesario conocer las reglas del negocio
- Las restricciones son normativas
- No se puede automatizar, es diferente en cada proceso, en cada organización
- La dependencia funcional es una restricción entre una o más columnas de una tabla



## Notación

- Clasificación:  $X \rightarrow Y$
- $X$  (funcionalmente) determina  $Y$
- **Para cada valor  $X$ , hay como máximo un valor  $Y$**
- $\text{StdNo} \rightarrow \text{StdCiudad}$  si cada valor de  $\text{stdNo}$  tiene como máximo un valor  $\text{StdCiudad}$
- $X$ : determinante



# Ejemplo



<u>StdNo</u>	StdClas	<u>ProgNo</u>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Ejemplo ProgNo -> ProgAnio
  - Hay como máximo un año para cada programación
  - Colocar ProgNo y ProgAnio en la misma tabla: ProgNo es único
- Ejemplo ProgNo, StdNo -> ProgNota
  - Hay como máximo una nota para cada combinación de ProgNo y StdNo
  - Coloque ProgNo y StdNo en la misma tabla con ProgNota: <ProgNo, StdNo> es único

# Ejemplo: encontrando dependencias funcionales



<b>StdNo</b>	StdClas	<b>ProgNo</b>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Primero analicemos las columnas que forman la clave primaria (individualmente) (analizamos contra todas las columnas)
- **StdNo** → StdClas :SI      **StdNo** → ProgAnio :NO      **StdNo** → ProgNota :NO
- **StdNo** → CursoNo :NO      **StdNo** → CursoDesc :NO
- Solo queda como dependencia funcional **SrtNo→StdClas** por lo que ya no analizaremos esa columna (esos dos deben ir juntos en la misma tabla)

# Ejemplo: encontrando dependencias funcionales



<b>StdNo</b>	StdClas	<b>ProgNo</b>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Analizamos la siguiente columna de la clave primaria
- **ProgNo** → ProgAnio :SI **ProgNo** → ProgNota :NO, **ProgNo** → CursoNo :SI, **ProgNo** → CursoDesc :SI
- Queda como dependencia funcional lo siguiente:
- **ProgNo** → **ProgAnio, CursoNo, CursoDesc** (deben ir juntos en una tabla)

# Ejemplo: encontrando dependencias funcionales



StdNo	StdClas	ProgNo	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Como ya acabamos las columnas individuales ahora analicemos la combinación de ambas contra las columnas que aun no tienen asignada una dependencia funcional
- **StdNo,ProgNo** → ProgNota :SI
- Queda como dependencia funcional lo siguiente:
- **StdNo,ProgNo** → **ProgNota** (deben ir juntos en una tabla)

# Ejemplo: encontrando dependencias funcionales



<b>StdNo</b>	StdClas	<b>ProgNo</b>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Las dependencias funcionales identificadas hasta el momento (las que deben ir en la misma tabla) son:
- **SrtNo** → **StdClas**
- **ProgNo** → **ProgAnio, CursoNo, CursoDesc**
- **StdNo, ProgNo** → **ProgNota**

# Ejemplo: encontrando dependencias funcionales



<b>StdNo</b>	StdClas	<b>ProgNo</b>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- **ProgNo  $\rightarrow$  ProgAnio, CursoNo, CursoDesc**
- ¿tendrá dependencias funcionales entre sus atributos (dependencias transitivas)?
- ProgAnio  $\rightarrow$  CursoNo :NO, ProgAnio  $\rightarrow$  CursoDesc :NO
- CursoNo  $\rightarrow$  CursoDesc :SI Deben ir en la misma tabla (por lo que se saca un atributo de la dependencia funcional)
- **ProgNo  $\rightarrow$  ProgAnio, CursoNo**
- **CursoNo  $\rightarrow$  CursoDesc**

# Ejemplo: encontrando dependencias funcionales



<b>StdNo</b>	StdClas	<b>ProgNo</b>	ProgAnio	ProgNota	CursoNo	CursoDesc
S1	JUN	O1	2013	3.5	C1	DB
S1	JUN	O2	2013	3.3	C2	VB
S2	JUN	O3	2014	3.1	C3	OO
S2	JUN	O2	2013	3.4	C2	VB

- Al final las tablas resultantes (dependencias funcionales y transitivas) :
- **SrtNo**→**StdClas**
- **ProgNo** → **ProgAnio, CursoNo**
- **CursoNo** → **CursoDesc**
- **StdNo,ProgNo** → **ProgNota**



# **Actividad de Clase**





# ACTIVIDAD DE CLASE

## CASO: TALLER DE REPARACIÓN DE RADIOS

Se desea controlar la actividad de reparación de radios en los talleres dedicados a esta tarea.

Cada usuario puede llevar diferentes radios para ser reparados, aunque un radio sólo pertenece a un usuario.

Para cada radio se conoce el número de la solicitud de reparación, que lo identifica, tipo de radio, descripción y tipo de rotura (**sencillas** o **complejas**).

De cada usuario se conoce su DNI, nombre y dirección.

Cada técnico de reparaciones pertenece a un taller y tiene un código, un nombre, una categoría (**A, B, C**) y un salario. En un taller laboran muchos técnicos de reparaciones.

De cada taller se conoce su código, que lo identifica, nombre y dirección. Los radios con roturas complejas sólo podrán ser atendidos por técnicos de categoría **A**, mientras que los radios con roturas sencillas pueden ser atendidos por cualquier técnico.

En cualquier caso, un técnico puede reparar muchos radios, pero un radio es reparado por un solo técnico.



# ACTIVIDAD DE CLASE

## Caso: ESCUELA PRIMARIA

Para el control de las escuelas primarias en la ciudad de Lima se tiene la siguiente información:

- De cada escuela, un número que la identifica, su nombre y su dirección. De cada aula, el número que la identifica, la cantidad de asientos que tiene y el piso en que se encuentra situada. De cada grupo de clases, un identificador del grupo, el grado escolar del grupo y la cantidad de alumnos que tiene. De cada maestro, su DNI, su nombre, su sexo y el año en que se graduó. De cada alumno, su número de expediente, su nombre, su sexo y su fecha de nacimiento.
- Una escuela tiene muchas aulas y muchos grupos de clases, pero cada aula pertenece a una escuela y lo mismo sucede con cada grupo.
- Un grupo siempre recibe clases en la misma aula y una aula pertenece a un solo grupo.
- En un grupo imparte clases un maestro y éste sólo imparte clases en un grupo. En cada grupo de clases hay muchos alumnos, pero un alumno forma parte de un solo grupo.



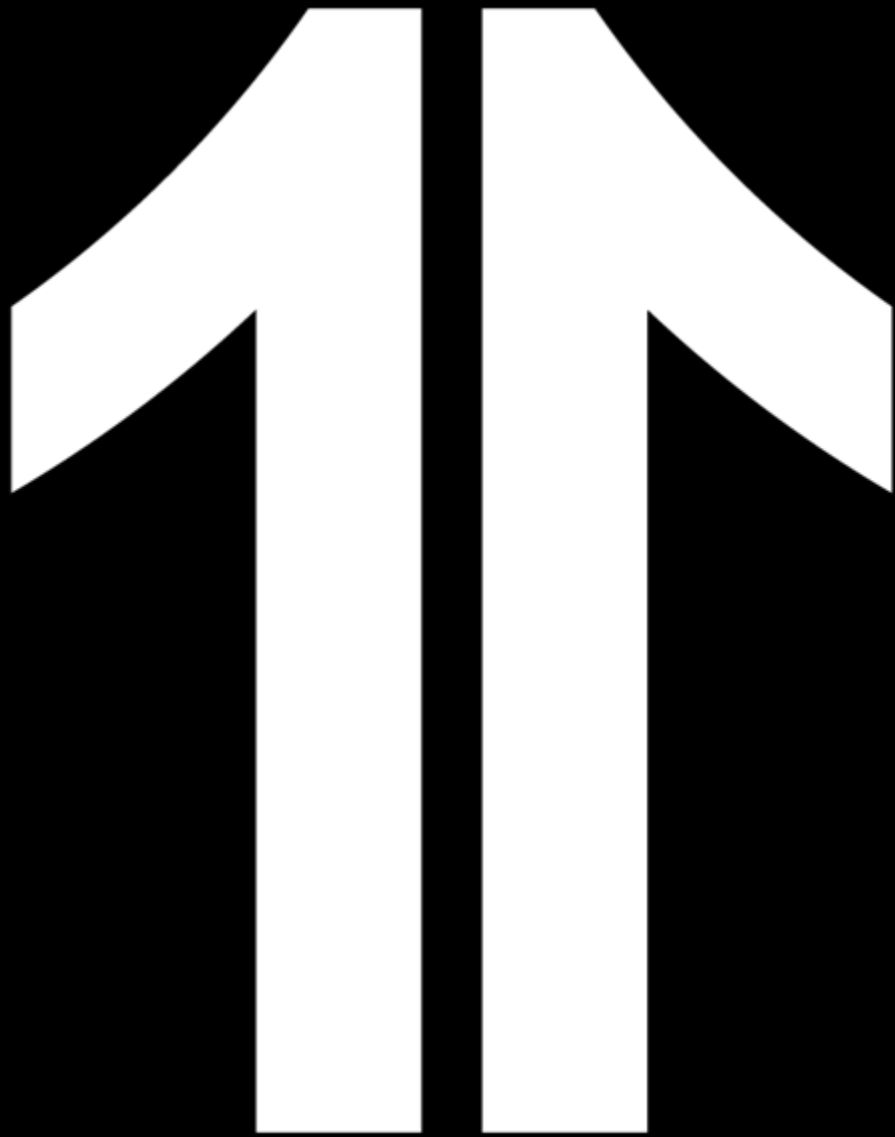
# CONCLUSIONES

1. El modelo entidad-relación es una herramienta fundamental en el diseño de bases de datos, ya que permite representar de manera visual y estructurada la información y las relaciones entre diferentes entidades. Esto facilita la comprensión del sistema que se está modelando.
2. Utilizando diagramas E-R, los diseñadores pueden identificar claramente las entidades relevantes, sus atributos y las relaciones que existen entre ellas. Esto ayuda a evitar ambigüedades y confusiones en la interpretación de los datos.
3. Los diagramas E-R sirven como un lenguaje común entre los diseñadores de bases de datos, desarrolladores y partes interesadas. Esto mejora la comunicación y asegura que todos los involucrados tengan una comprensión clara del modelo de datos.



# REFERENCIAS BIBLIOGRÁFICAS

1. Hernández, M. A., & Rodríguez, J. A. (2020). MODELOS DE LOS DATOS. En este documento se describen los modelos de datos, incluyendo el modelo entidad-relación y su aplicación en el diseño de bases de datos. [Enlace](#).
2. DANE. (n.d.). Recomendaciones para elaborar modelos entidad-relación. Este documento proporciona pautas y consideraciones para la creación de modelos entidad-relación, así como la notación recomendada para su diseño. [Enlace](#).
3. Saltón, G., & McGill, M. J. (1983). Introduction to Modern Information Retrieval. Este libro es una referencia clásica que incluye conceptos sobre el diseño de bases de datos y el modelo entidad-relación, abordando su importancia en la recuperación de información. [Enlace](#).



**GRACIAS**

