



códigofacilito



SQL

Diseño y normalización de base de datos

Mayra - Ingeniera en Sistemas





>_ Diagrama Entidad Relación

>_ Normalización de bases de datos





¿Que es un diagrama Entidad Relación?

Un modelo entidad-relación es una herramienta para el modelo de datos, la cual facilita la representación de entidades de una base de datos. Los diagramas ER son un lenguaje gráfico para describir conceptos.

Entidad

Las entidades son el fundamento del modelo entidad relación. Podemos adoptar como definición de entidad cualquier cosa o parte del mundo que es distinguible del resto.

Atributos

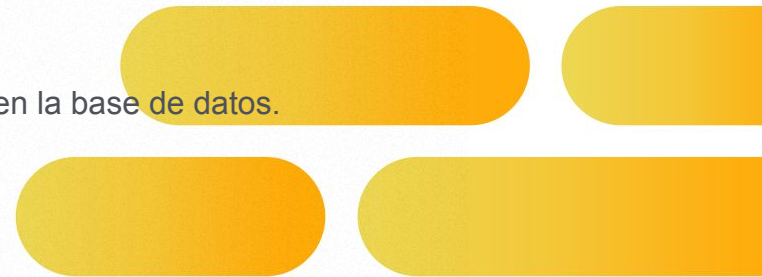
Los atributos son las características que definen o identifican a una entidad. Éstas pueden ser muchas, y el diseñador sólo utiliza o implementa las que considere más relevantes.

Relación

Describe cierta dependencia entre entidades o permite la asociación de las mismas.

Restricciones

Son reglas que deben respetar las entidades y relaciones almacenadas en la base de datos.

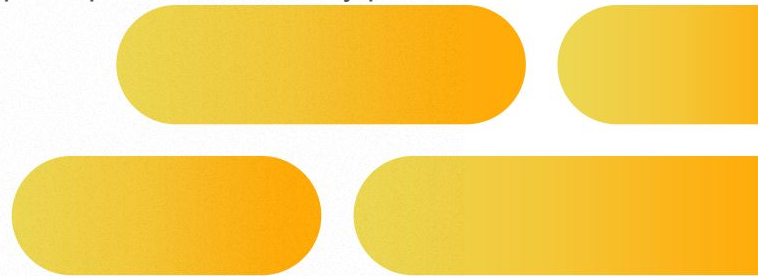




Cardinalidad

La cardinalidad de la correspondencia indica el número de entidades con las que puede estar relacionada una entidad dada.

- **Uno a Uno: (1:1)** Un registro de una entidad A se relaciona con solo un registro en una entidad B.
- **Uno a Muchos: (1:N)** Un registro en una entidad en A se relaciona con uno o muchos registros en una entidad B. Pero los registros de B solamente se relacionan con un registro en A.
- **Muchos a Muchos: (N:M)** Una entidad en A se puede relacionar con 1 o con muchas entidades en B y viceversa.
- **"0"** si cada instancia de la entidad no está obligada a participar en la relación.
- **"1"** si toda instancia de la entidad está obligada a participar en la relación y, además, solamente participa una vez.
- **"N"**, **"M"**, ó **"*"** si cada instancia de la entidad no está obligada a participar en la relación y puede hacerlo cualquier número de veces.





Tipos de Diagrama Entidad Relación





Diagrama ER - Modelo Conceptual

- Incluye las entidades importantes y las relaciones entre ellas.
- No se especifica ningún atributo.
- No se especifica ninguna clave principal.

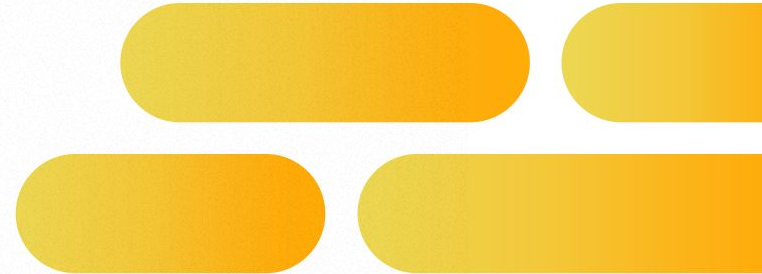
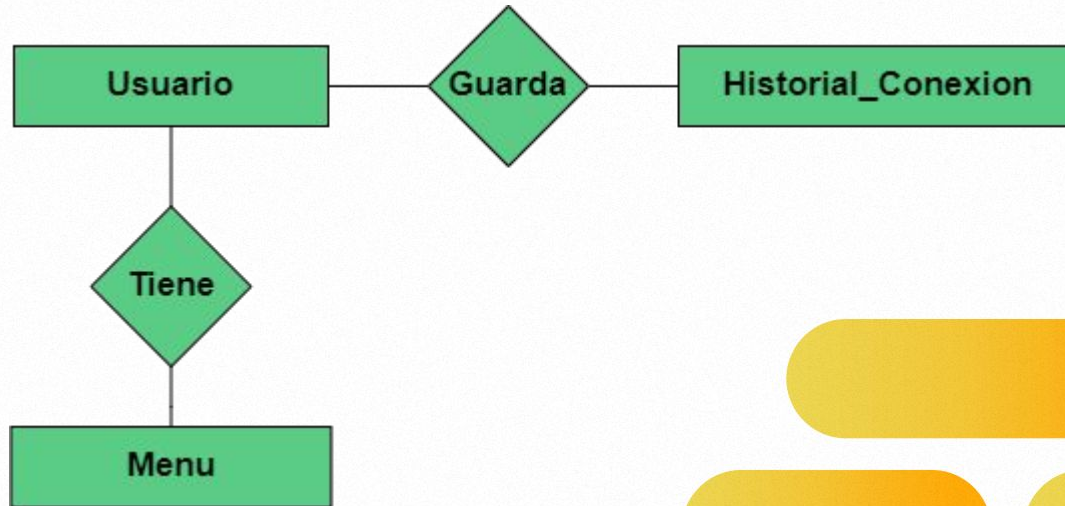




Diagrama ER - Modelo Lógico

- Incluye todas las entidades y relaciones entre ellos.
- Todos los atributos para cada entidad están especificados.
- La clave principal para cada entidad está especificada.
- Se especifican las claves externas (claves que identifican la relación entre diferentes entidades).
- La normalización ocurre en este nivel.

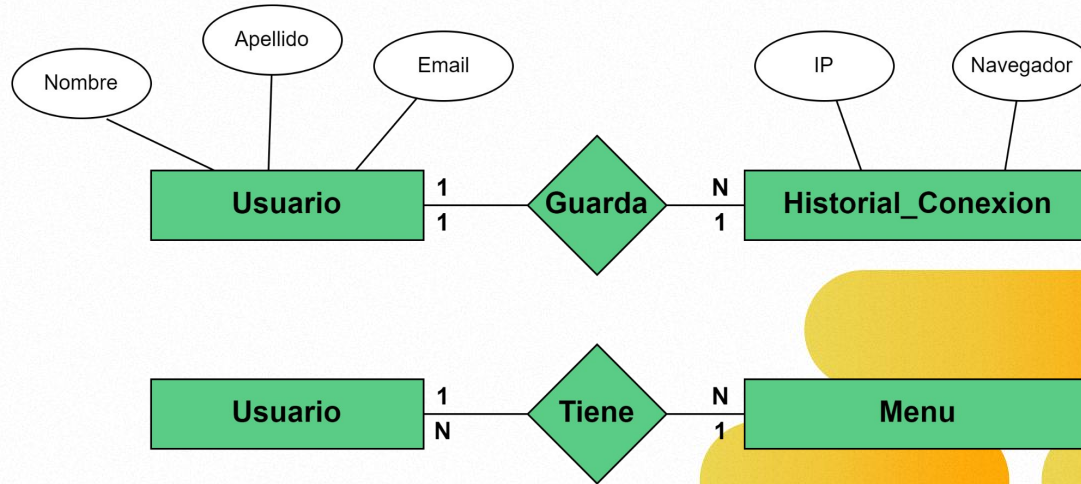
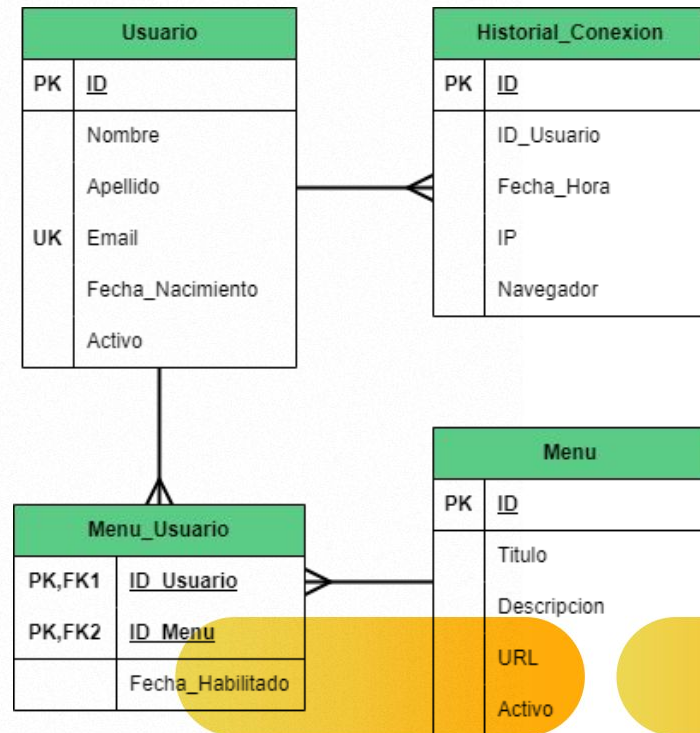




Diagrama ER - Modelo Físico

- Especificación de todas las tablas y columnas.
- Las claves externas se usan para identificar relaciones entre tablas.
- La desnormalización puede ocurrir según los requisitos del usuario.

[Este tipo de Diagrama ER es el que se utiliza comúnmente en las empresas.](#)





> Entidad o Tabla:

Propiedades:

- Nombre de la tabla, debe ser único
- Campos o atributos
- Cada campo tiene un tipo de dato asociado. (El tipo de dato no siempre se especifica)
- La tabla puede tener una llave primaria. (PK)
Si ningún campo de la tabla es único, se puede usar un correlativo. (Lo recomendable es que todas las tablas tengan llave primaria)
- Una tabla puede tener una o varias llaves foráneas si hubieran relaciones de uno a muchos. (FK)
- Los campos pueden tener otros constraints como Unique Key (UK), Not Null, etc.

Usuario	
PK	<u>ID (INT)</u>
UK	Nombre (VARCHAR)
	Apellido (VARCHAR)
	Email (VARCHAR)
	Fecha_Nacimiento (DATETIME)
	Activo (TINYINT)

Usuario	
PK	<u>ID</u>
UK	Nombre
	Apellido
	Email
	Fecha_Nacimiento
	Activo





>_ Llaves Primarias:

- Puede estar conformada por uno o varios campos.
- Ayudan a cuidar que no haya duplicados.
- Lo más común es que las llaves primarias sean correlativos (automáticos o manuales) pero cualquier campo podría ser llave primaria siempre y cuando sea único null.

UsuarioID	Usuario	Nombre	Genero
1	@maycode	Mayra	Female
2	@mayramorataya	Mayra	Female
3	@mayra_morataya	Mayra	Female

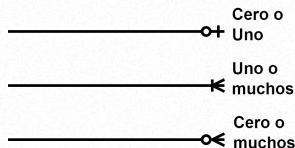
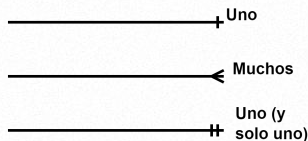
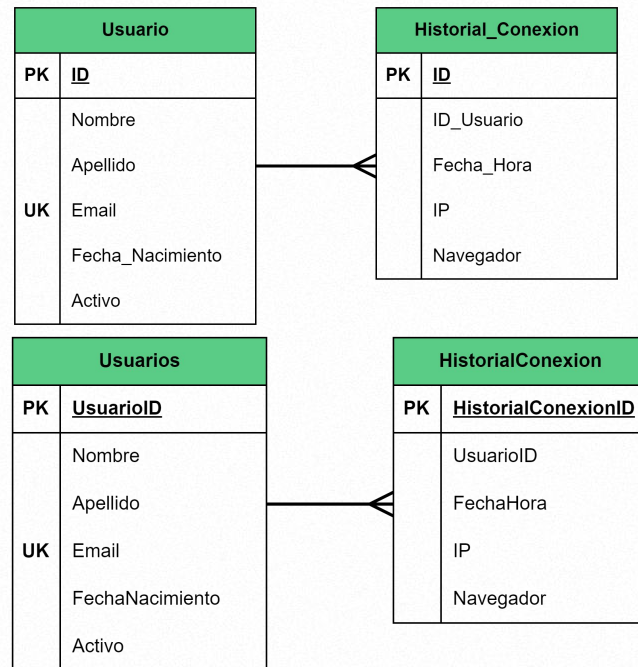
Usuario	Nombre	Genero
@maycode	Mayra	Female
@mayramorataya	Mayra	Female
@mayra_morataya	Mayra	Female





>_ Llaves Foráneas:

- Puede estar conformada por uno o varios campos.
- Es utilizada para conectar los datos de una tabla con otra(s).
- La llave primaria de una tabla pasa a ser llave foránea de otra tabla.
- El campo (o los campos) que son llave primaria en una tabla pueden tener el mismo nombre o no, en la tabla en la que son llave foránea.
- El campo (o los campos) que son llave primaria en una tabla y pasan a ser llave foránea a otra deben de tener el mismo tipo de dato.





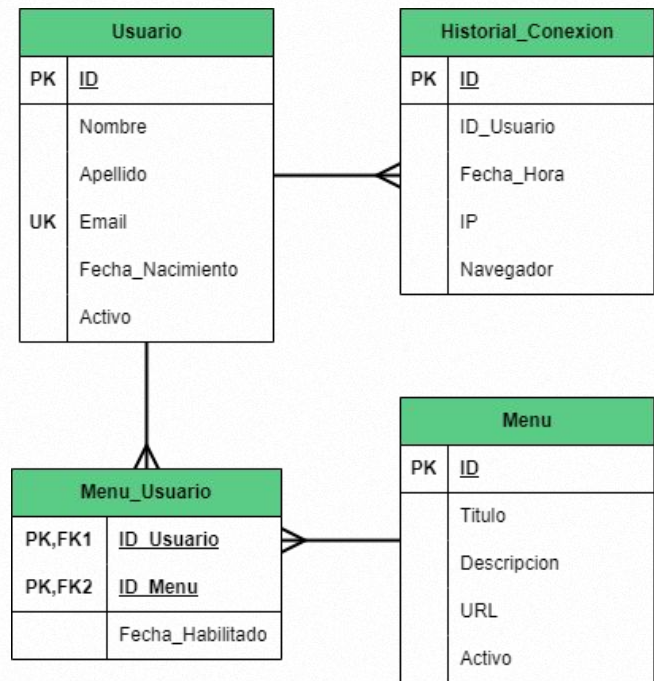
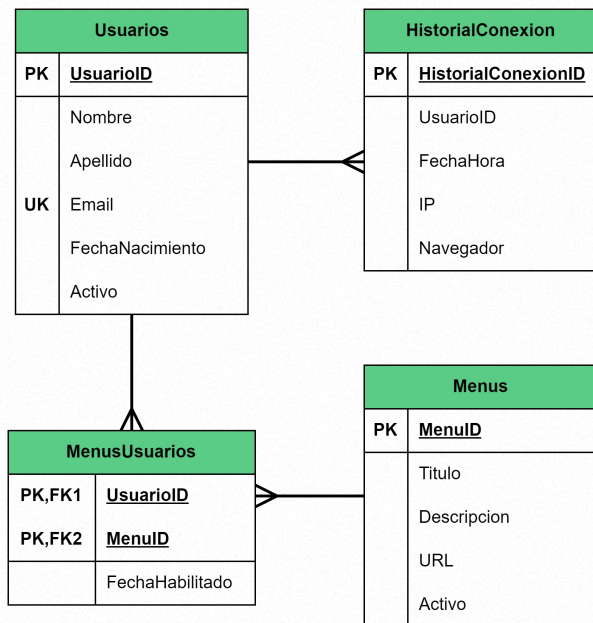
Tips al diagramar en el trabajo





> Observar el diagrama ER actual de la BD de la empresa, y lo nuevo que se diagrame que siga el mismo patrón.

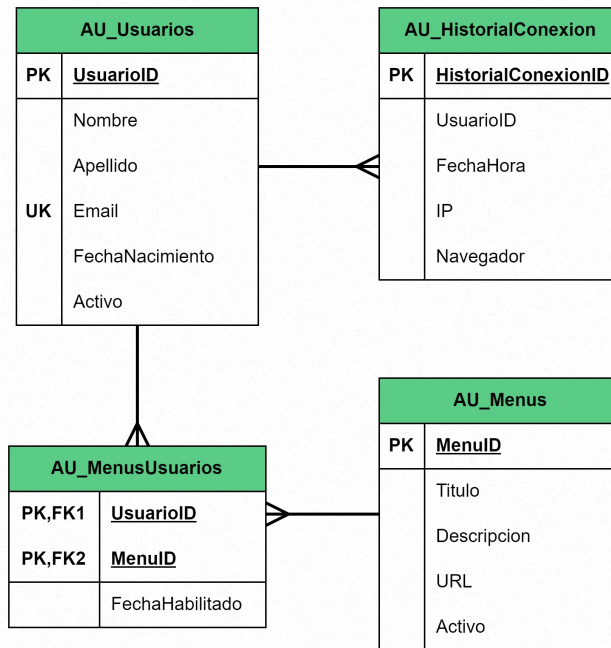
Observar la nomenclatura de nombres de objetos (tablas, campos, triggers, funciones, etc) y replicarla a los nuevos objetos.





- > Observar si la empresa utiliza mnemónicos, o si es una BD completamente nueva, decidir desde el inicio si será conveniente o no usar mnemónicos.

VE_Ventas	
PK	<u>Venta_ID</u>
	FechaVenta
	Descripcion
	...





>_ Utilizar nombres de objetos en el mismo idioma (todo español o todo inglés).

User	
PK	ID
UK	FirstName
	LastName
	Email
	BirthDate
	Active

ConnectionLog	
PK	ID
	UserID
	DateTime
	IP
	Browser

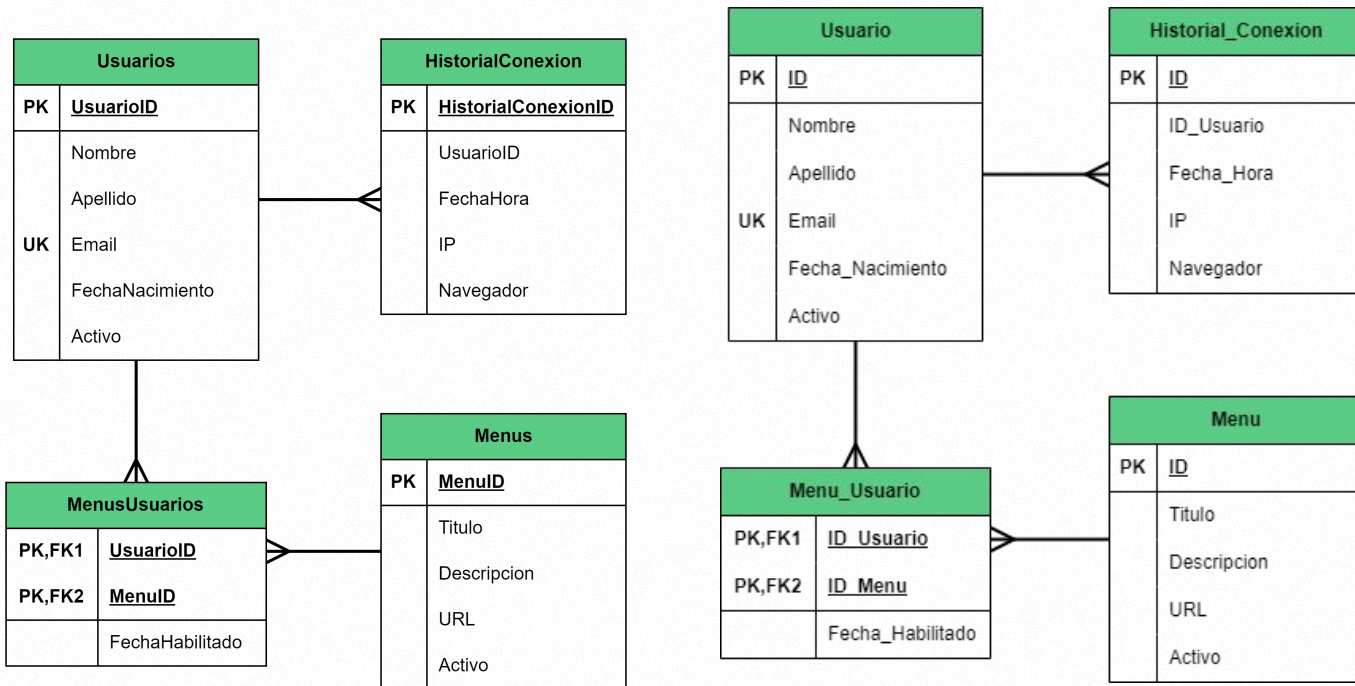
Usuario	
PK	ID
UK	Nombre
	Apellido
	Email
	Fecha_Nacimiento
	Activo

Historial_Conexion	
PK	ID
	ID_Usuario
	Fecha_Hora
	IP
	Navegador





- > Utilizar todos los nombres de tablas en plural o todos en singular.
No hay una regla estricta, la práctica más común y recomendada es utilizar nombres de tablas en singular.





>_ Otros Ejemplos:

- Para nombres compuestos por varias palabras definir si se usará algún caracter como _ para separarlas o no.
- Para los nombres muy largos de tablas y/o campos se usarán abreviaturas o no.
- Algunos gestores de BD permiten definir nombres de objetos con caracteres especiales, definir si se usarán o no.
- Para campos con 2 posibles valores como True / False, 1 / 0, S / N, Y / N, definir con qué tipo de dato se almacenarán para aplicar la misma regla a todas las tablas.
- Para tablas y campos que guardan status (En Proceso, Exitoso, Fallido, etc) definir si sus llaves primarias serán letras P = En proceso, E = Éxito, F = Fallido o correlativos.

Es muy importante que la base de datos sea homogénea, es decir que todos los nombres de objetos (tablas, campos, funciones, etc) tengan una misma nomenclatura para facilitar el mantenimiento y hacer queries.





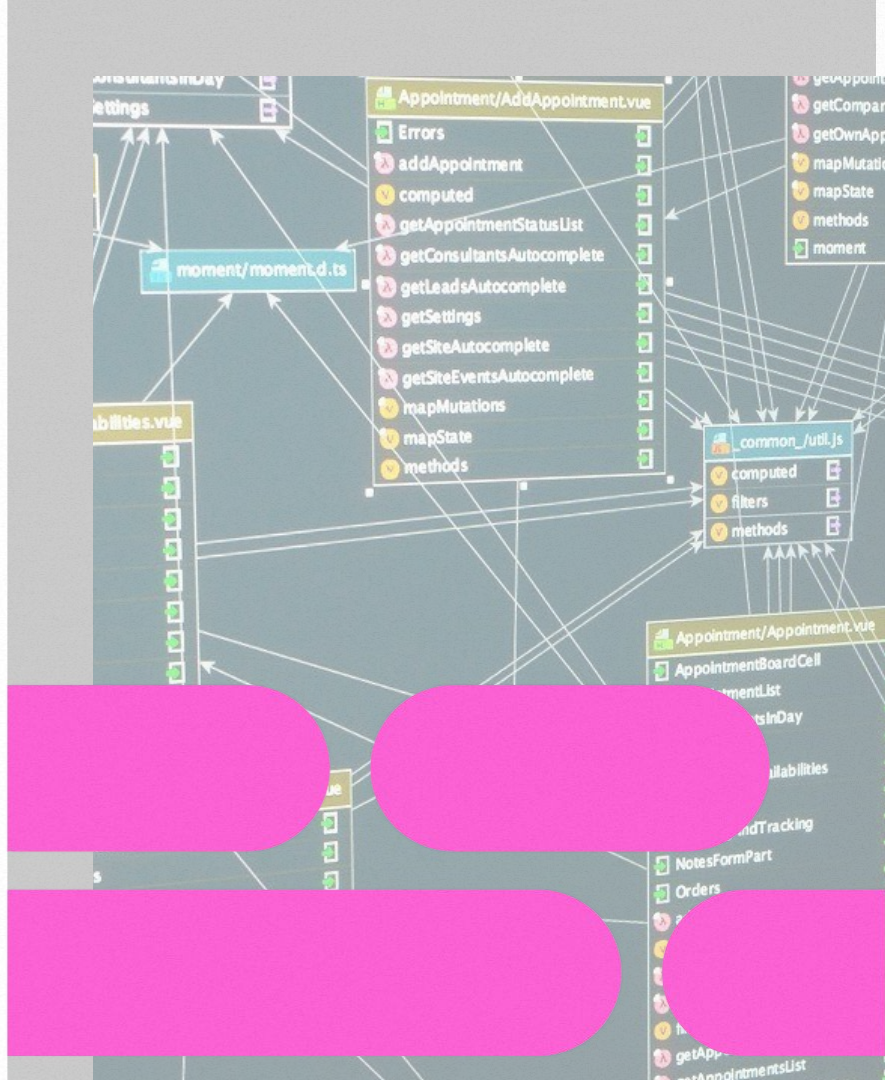
Normalización de bases de datos





> ¿Que es la normalización de bases de datos?

Es un proceso importante en el diseño de bases de datos relacionales que consiste en designar y **aplicar una serie de reglas** a las relaciones obtenidas tras el paso del modelo entidad-relación **al modelo relacional con objeto de minimizar la redundancia de datos**, facilitando su gestión posterior.



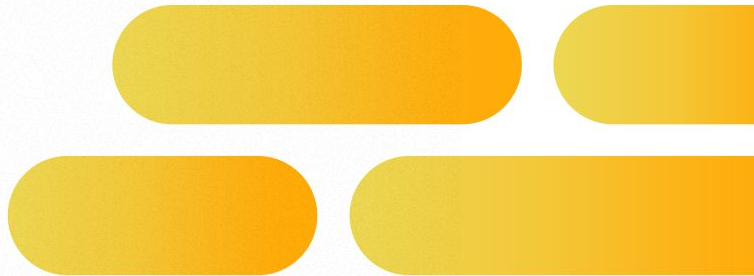


Aspectos importantes

El proceso de normalizar la BD es básicamente reducir la redundancia.
El diseño debe ser lógico o acorde a las necesidades del usuario.

Al diseñar una BD hazte estas preguntas:

- ¿Que tipo de datos necesito guardar?
- ¿Cómo van a acceder los usuarios a los datos?
- ¿Los usuarios tendrán privilegios especiales sobre los datos?
- ¿Hay alguna forma de agrupar los datos?
- ¿Cómo se relacionan los datos entre ellos?
- ¿Cómo asegurar la integridad de los datos?
- ¿Los datos se repiten?





Convención de nombres

Es importante elegir nombres relevantes a la información que allí se almacenará de esa forma serán fáciles de recordar.

Esto ayuda a mantener los objetos ordenados y evitar confusiones.





Beneficios de la normalización

- Mantiene la BD organizada y fácil de usar.
- Reduce la repetición y redundancia de la información.
- Mejora la seguridad para el sistema y usuarios.
- Mejora la flexibilidad en el diseño.
- Asegura la consistencia.

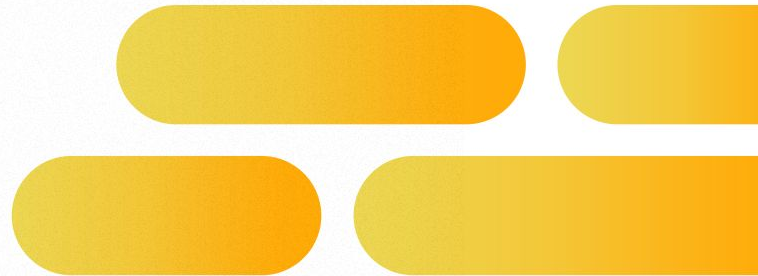




Contras de la Normalización

Puede reducir el rendimiento de la base de datos en algunos casos, eso es porque se requiere más poder de cómputo y memoria para realizar las tareas.

Una base de datos normalizada requiere unir muchos datos y tablas para devolver el resultado deseado.

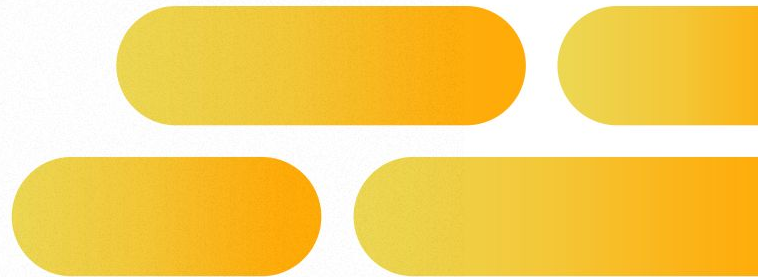




Desnormalización

Consiste en tomar una BD normalizada y modificarla para aceptar redundancia. Este proceso puede llegar a ser útil cuando se desea mejorar el rendimiento.

Depende la situación o el caso de negocio, puede ser buena idea normalizar o desnormalizar.





>_ Formas Normales

Una forma normal es un método para identificar los niveles o profundidad que se requiere para normalizar la BD.

No todas las bases de datos se llegan a normalizar a la misma profundidad, siempre es importante tener en cuenta las necesidades del usuario a resolver.

Una base de datos puede llegar a tener hasta 6 formas normales.

Las formas normales se aplican de forma secuencial, es decir primero la 1NF luego 2NF, etc.

[La mayoría de bases de datos reales en el mundo llegan hasta la tercera forma normal.](#)





Primera Forma Normal (**1NF**)





- >_ 1. Todas las columnas de las tablas contienen valores atómicos.

NombrePaciente	NombreDepartamento
Juan, Ana, Maria	Cardiologia
Marcos, Rodrigo	Medicina General

La columna NombrePaciente tiene muchos valores. ❌





>_ 2. No contiene grupos de columnas repetidas.

Se refiere a columnas consideradas repetidas, que genera desperdicio de espacio y problemas al mantener la base de datos.

NombrePaciente1	NombrePaciente2	NombrePaciente3	NombreDepartamento
Juan	Ana	Maria	Cardiologia
Marcos	Rodrigo		Medicina General

En el tiempo con este diseño se generaría el problema de crear nuevas columnas "NombrePaciente4" etc. ❌





>_ 2. Cada fila necesita un identificador único (Llave primaria).

NombrePaciente	NombreDepartamento
Juan, Ana, Maria	Cardiologia
Marcos, Rodrigo	Medicina General

Este diseño no tiene llave primaria. ✗





>_ Si una base de datos cumple todas las reglas anteriores se considera que está en la primera forma normal.

PacienteID	NombrePaciente	NombreDepartamento
1	Juana	Cardiologia
2	Ana	Cardiologia
3	Maria	Cardiologia
4	Marcos	Medicina General
5	Rodrigo	Medicina General





Segunda Forma Normal (**2NF**)





>_ 1. Cumple todas las reglas de la 1NF.

PacienteID	NombrePaciente	NombreDepartamento
1	Juana	Cardiologia
2	Ana	Cardiologia
3	Maria	Cardiologia
4	Marcos	Medicina General
5	Rodrigo	Medicina General





> 2. No hay datos redundantes en ninguna columna, excepto las llaves foráneas.

Deben crearse tablas nuevas para columnas con valores redundantes. Y como consecuencia crear las relaciones entre las tablas.

PacienteID	NombrePaciente	NombreDepartamento
1	Juana	Cardiologia
2	Ana	Cardiologia
3	Maria	Cardiologia
4	Marcos	Medicina General
5	Rodrigo	Medicina General

Hay valores repetidos en la columna NombreDepartamento. ✖





>_ Si una base de datos cumple todas las reglas anteriores se considera que está en la segunda forma normal.

PacienteID	NombrePaciente	DepartamentoID	DepartamentoID	NombreDepartamento
1	Juana	1	1	Cardiologia
2	Ana	1	2	Medicina General
3	Maria	1		
4	Marcos	2		
5	Rodrigo	2		





Tercera Forma Normal (**3NF**)





>_ 1. Cumple todas las reglas de la 2NF.





> 2. Todas las columnas en las tablas dependen completamente de llaves primarias.

PacienteID	NombrePaciente	Edad	Genero	NombreDepartamento	JefeDepartamento	NoCamas
1	Juana	45	Female	Cardiologia	Dr. Hernandez	100
2	Ana	54	Female	Cardiologia	Dra. Martinez	140
3	Maria	39	Female	Cardiologia	Dr. Perez	120
4	Marcos	34	Male	Medicina General	Dr. Alvarez	100
5	Rodrigo	71	Male	Medicina General	Dra. Garcia	123

Todos los datos del departamento dependen del PacienteID. ❌





> Si una base de datos cumple todas las reglas anteriores se considera que está en la tercera forma normal.

DepartamentoID	NombreDepartamen	JefeDepartamento	NoCamas
1	Cardiologia	Dr. Hernandez	140
2	Medicina General	Dr. Alvarez	100

PacienteID	NombrePaciente	Edad	Genero	DepartamentoID
1	Juana	45	Female	1
2	Ana	54	Female	1
3	Maria	39	Female	1
4	Marcos	34	Male	2
5	Rodrigo	71	Male	2



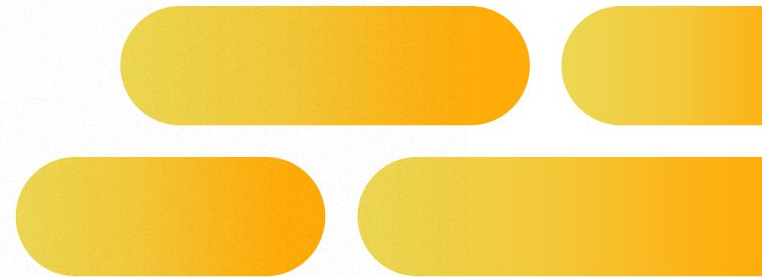


Forma normal de Boyce-Codd (**BCNF**)

También conocida como Forma Normal 3.5

Es considera una versión más estricta de la 3NF.

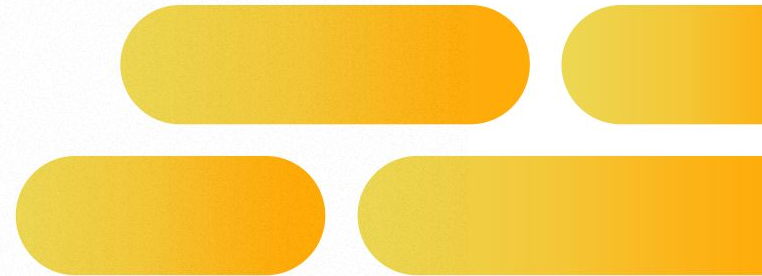
Los registros de la tabla deben ser considerados únicos, se considera único basado en la combinación de las columnas.





Cuarta Forma Normal (4NF)

1. Cumple todas las reglas anteriores.
2. Es necesario dividir las tablas en múltiples tablas más pequeñas, de modo que cada tabla tenga una relación de uno a uno con la clave primaria. Así se evitan las redundancias y las anomalías de actualización que pueden surgir debido a las dependencias multivaluadas.





Quinta Forma Normal (5NF)

1. Cumple todas las reglas anteriores.
2. Elimina dependencias de unión y dependencias de unión multivaluadas dividiendo las tablas en entidades separadas.
Cuando una tabla contiene información sobre dos o más entidades independientes, se pueden dividir en tablas separadas para evitar dependencias no deseadas.

