**UNIDAD 2: Características de Base de Datos**

1. [UNIDAD 2: Introducción](https://learn.nextu.com/mod/page/view.php?id=9657&pid=P_WEB_DATABASE)

# **Tipos de Bases de Datos**

Existen diversos tipos de bases de datos, los cuales se han agrupado en dos grandes categorías, las **bases de datos relacionales** y las **no relacionales**. Las primeras representan la forma más tradicional de organizar los datos en estructuras, generalmente, conocidas como **tablas**. La segunda engloba todas aquellas formas de constituir datos de una manera **flexible**, que en algunos casos puede ser una alternativa más natural de resolver los problemas. En esta unidad, nos adentramos un poco más en el mundo de las bases de datos, permitiéndote así reconocer estas dos categorías en función de sus **componentes estructurales**. Usando un enfoque más **práctico**, te daremos la oportunidad de aplicar los elementos de las bases de datos SQL y NoSQL en diversos escenarios, con la finalidad de que puedas diferenciar en qué situaciones un paradigma puede resultar mejor que el otro.

# **Objetivos de aprendizaje**

1. **Usar las sentencias** básicas de SQL en la gestión de bases de datos relacionales.
2. **Diferenciar las bases de datos** SQL de las NoSQL en función de sus características elementales.
3. **Identificar la estructura** de los documentos JSON en función de los valores, objetos y arreglos.
4. Lección 1: Tipos de Base de Datos

[2.1. SQL vs NoSQL](https://learn.nextu.com/mod/lesson/view.php?id=9658&pid=P_WEB_DATABASE)

SQL – Relacionales:

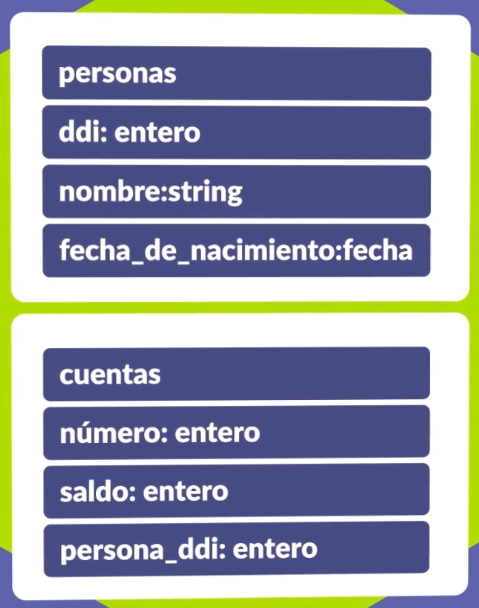
* Los datos se estructuran en un esquema, como se van a organizar las entidades, en tablas, todos los registros contendrán la misma estructura, sino se indica un valor en una columna se tomará un valor por defecto que en la mayoría de los casos será nulo.
* Lenguaje manejador de la base de datos
* Se basan en el principio de integridad que significa que los datos no deben ser redundantes, los datos siempre deben ser consistentes
* Se caracteriza por poseer una escalabilidad Vertical, lo cual quiere decir que si se desea una DB más potente solo se debe incrementar el tamaño del recurso, si necesitamos más capacidad para la base de datos significa que necesitamos más capacidad en los servidores.
* SQL más conocidos: MySQL, PostgreSQL, Oracle, Microsoft Access

NoSQL – No Relacionales

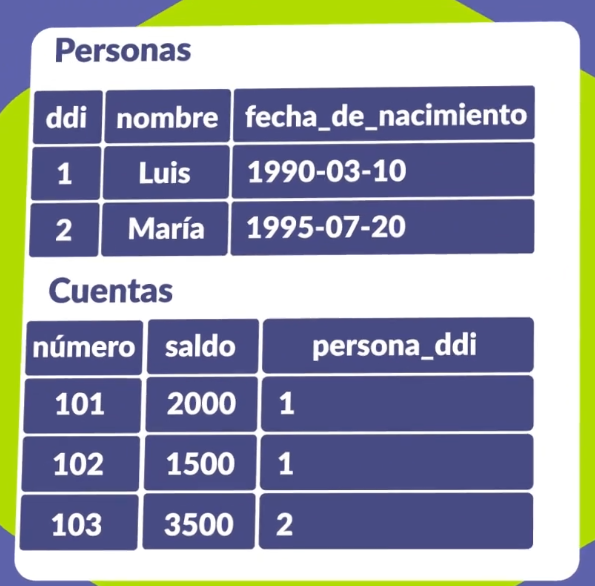
* Hay diversas alternativas para almacenar los datos, no poseen una estructura definida
* Alta velocidad de acceso y alta disponibilidad de los datos, incluso permitiendo mucha redundancia
* Las bases de datos se agrupan en nodos o clusters, así si deseamos aumentar la capacidad de la base de datos solamente se aumentan la cantidad de nodos
* Existen varios de base de datos no relacionales:
  + Base de datos orientada a documentos, como: MongoDB y CouchDB
  + Base de datos orientada a clave/valor, como MemCacheD y Redis
  + Base de datos orientada a multicolumna, como: Cassandra
  + Base de datos orientada a grafos, como: Neo4J

Ejemplo de DB para un Banco:

Tablas:



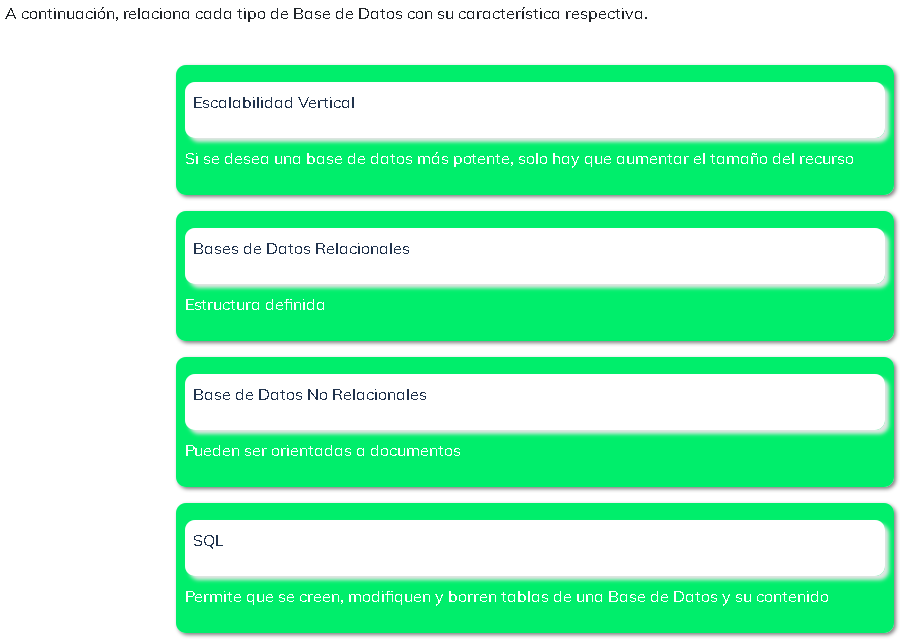
Datos: Se ha dado una relación de la tabla a través del campo persona\_ddi 🡪 ddi



2.2. Actividad Interactiva 1

# Analiza y relaciona

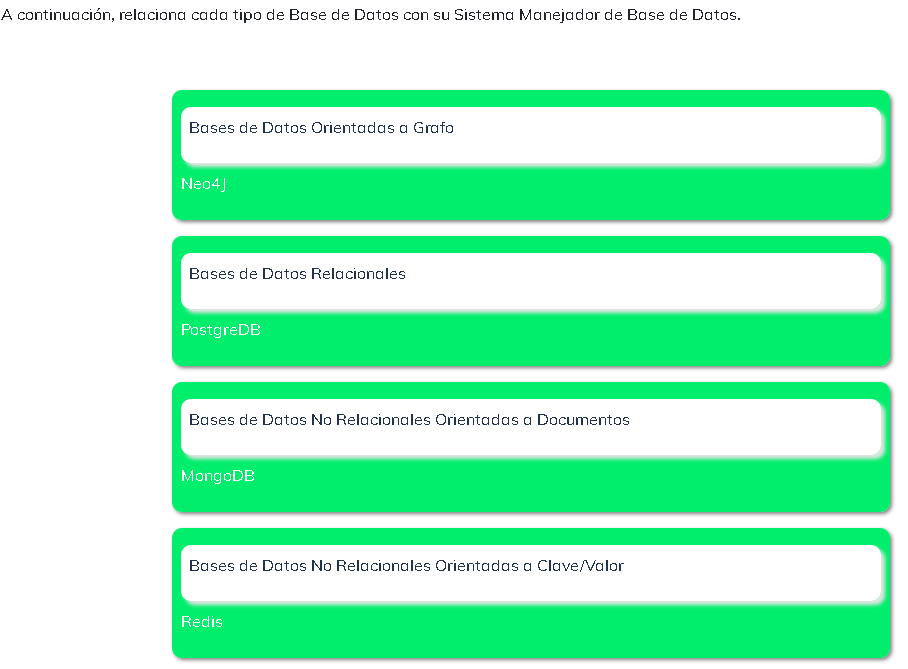
A continuación, relaciona cada tipo de Base de Datos con su característica respectiva.



2.3. Actividad Interactiva 2

# Analiza y relaciona

A continuación, relaciona cada tipo de Base de Datos con su Sistema Manejador de Base de Datos.



2.4. Ejercicio 1

Tenemos un sistema de manejo de las tareas de los estudiantes de una Universidad. Hemos identificado dos Entidades: Estudiante y Calificación. La Entidad Estudiante tiene tres Atributos: Identificador del Estudiante (ID) del tipo entero, Nombre del tipo string y Apellido del tipo string. La Entidad Calificación tiene como Atributos: Tarea del tipo string, Fecha de Entrega del tipo fecha, Calificación Obtenida del tipo entero y Estudiante ID que referencia al estudiante que realizó la tarea y es del tipo entero.

**Realiza la representación en tablas (usando estándar de nombres) de estos elementos de la base de datos usando el paradigma Relacional para los siguientes casos:**

* El estudiante de Luis (Nombre) Aguilar (Apellido) y ID (123) obtuvo la calificación de 50 puntos en la Tarea 1 entregada el 16 de marzo de 2018.
* La estudiante de Manuela (Nombre) Silva (Apellido) y ID (567) obtuvo la calificación de 90 puntos en la Tarea 1 entregada el 15 de marzo de 2018.

2.5. Más tipos NoSQL y Comparación

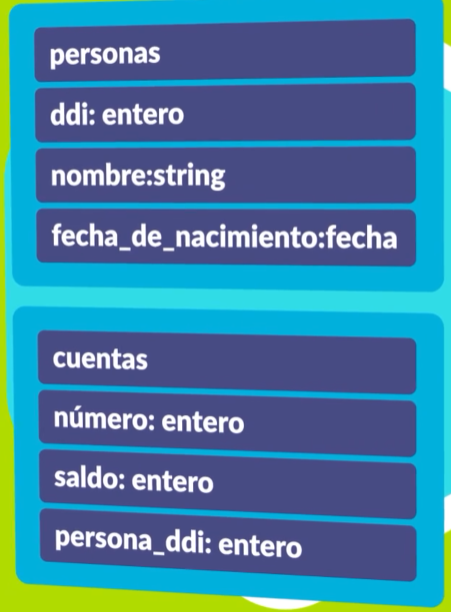
Hay mucha diversidad es la característica principal de las DB No relacionales.

Comparación:

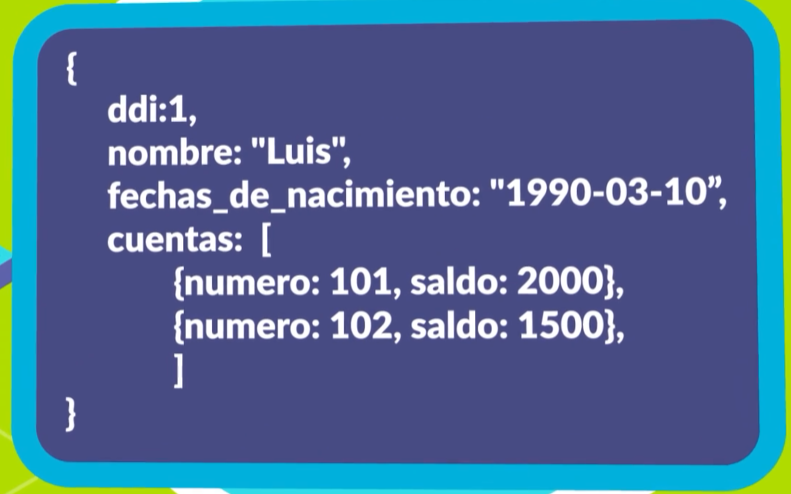
SQL

NoSQL – se representan en forma de documento, y un documento puede estar embebido en otro formando una jerarquía

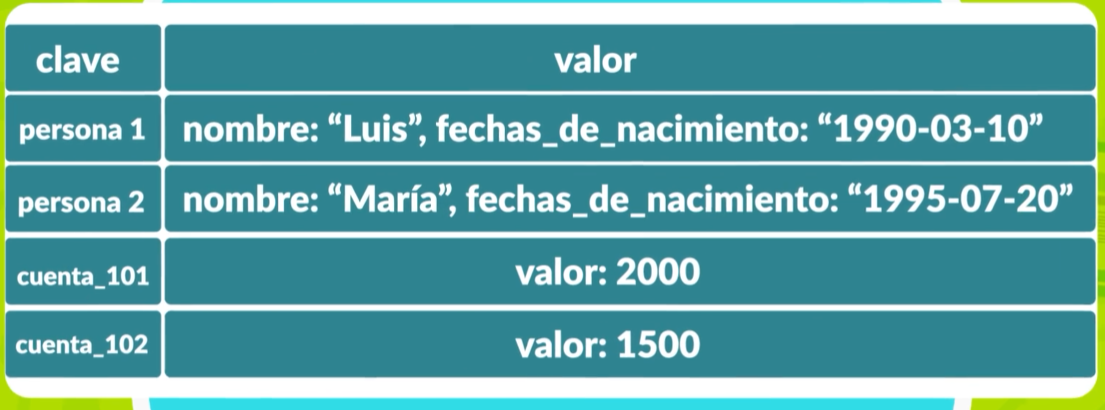
Ejemplo de cuenta bancaria



Los datos se almacenan en colecciones, las colecciones tienen documentos:



Las base de datos no relacionales orientadas a clave >> valor, es una forma más simple de guardar y recuperar los datos, esto hace que su implementación sea eficiente y la recuperación de los datos sea rápida. Usualmente estos sistemas trabajan con datos en memoria. Así que son ampliamente utilizados en sistemas que requieren almacenar datos en cache. Tales como el cache de páginas web.

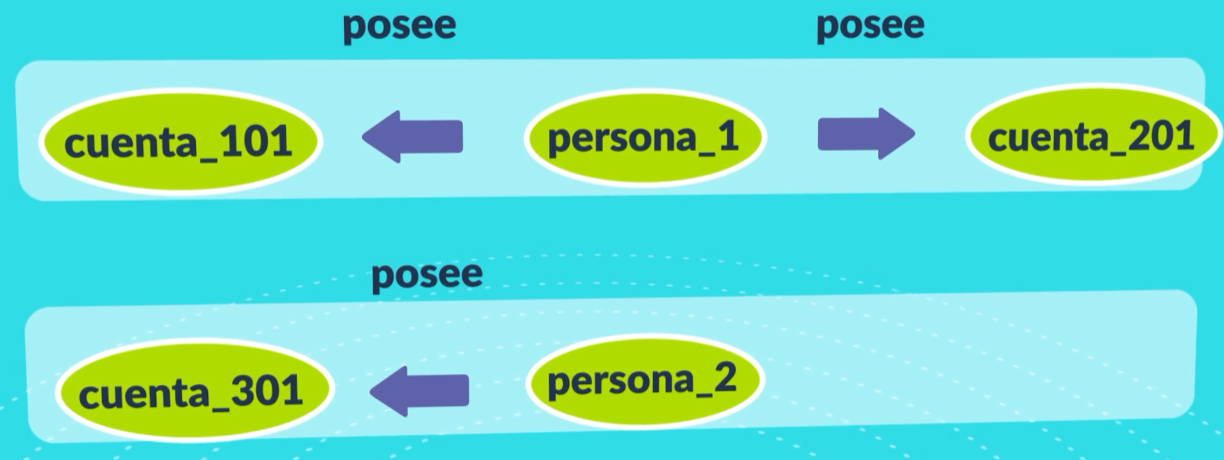


Proporciona una gran flexibilidad donde no se define una estructura de como funciona una persona o una cuenta. Dada una clave se da un valor, y para dada una clave recuperamos un valor.

El paradigma no relacional es el orientado a grafo, un grafo se define como un conjunto de nodos interconectados por arcos, que se usan para representar sistemas, tales como: una red de computadoras, donde los nodos son las computadoras y los arcos los enlaces de comunicación.

Las redes sociales también se pueden representar usando grafos, donde cada nodo es una persona, y los arcos muestran como se conectan a través del uso de alguna red social.

En la base de datos orientados a grafos vamos a encontrar nodos relacionados entre ellos. Los nodos a su vez van a tener propiedades o atributos.



Comparación:

SQL – relacionales

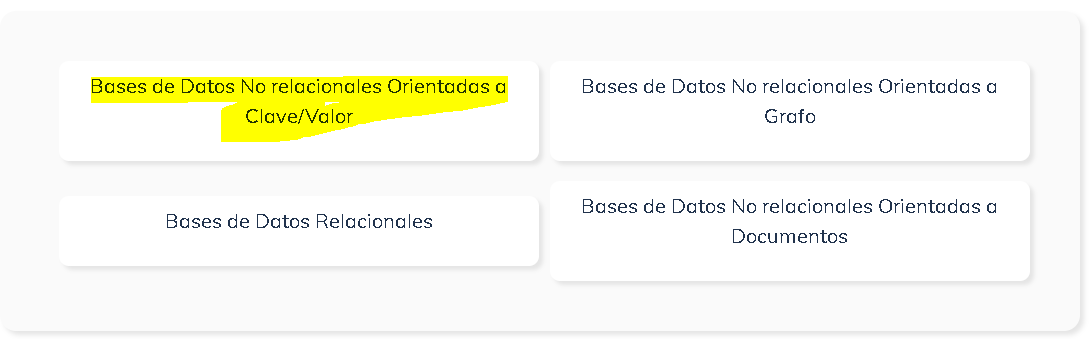
NoSQL – No relacionales

|  |  |  |
| --- | --- | --- |
| Criterio | SQL | NoSQL |
| Estructura | Definida (Definición de columnas y tablas y que tipo de datos) | Flexible, por ejemplo en orientada a documentos. Un documento puede tener 3 atributos y otro solamente 2. |
| Escalabilidad | Escalabilidad Vertical – Que se refiere a que tenemos que tener un servidor con más recursos si se requiere una base de datos con más capacidad | Escalabilidad Horizontal – que significa que está diseñadas para conectarse a otros nodos. Es decir que podemos agregar más nodos al sistema, así haciéndolo escalable |
| Orientación | Integridad / Consistencia - | Velocidad – Disponibilidad / Cluster |
| Acceso | SQL – Se hace uso del lenguaje SQL | Variable – se tiene una variedad de mecanismos para el acceso a los datos |

**2.6. Actividad Interactiva 3**

**Prueba tus conocimientos**

Usualmente las \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ trabajan con datos en memoria, así que son ampliamente usadas en aplicaciones que requieren almacenar datos en caché, tales como el caché de páginas Web.



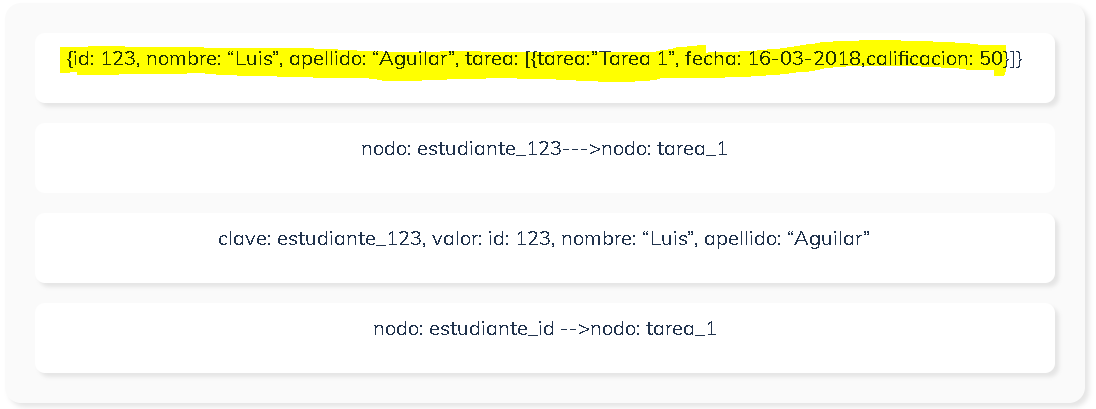
2.7. Actividad Interactiva 4

**Prueba tus conocimientos**

Tenemos un sistema de manejo de las tareas de los estudiantes de una Universidad. Hemos identificado dos Entidades: Estudiante y Calificación. La Entidad Estudiante tiene tres Atributos: Identificador del Estudiante (ID) del tipo entero, Nombre del tipo string y Apellido del tipo string. La Entidad Calificación tiene tres Atributos: Tarea del tipo string, Fecha de Entrega del tipo fecha, Calificación Obtenida del tipo entero y EstudianteID que referencia al estudiante que realizó la tarea y es del tipo entero. Considerando la representación en documento JSON de estos elementos de la base de datos usando el paradigma Relacional para el siguiente caso:

El estudiante de Luis (Nombre) Aguilar (Apellido) y ID (123) obtuvo la calificación de 50 puntos en la Tarea 1 entregada el 16 de marzo de 2018.

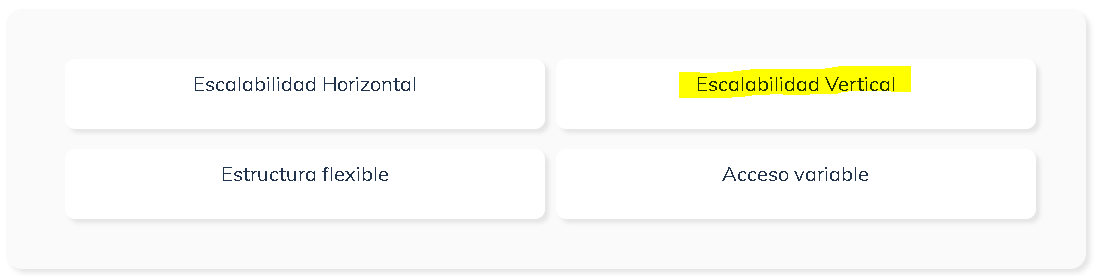
**¿Cuál de las siguientes representaciones corresponde a un sistema de base de datos No Relacional Orientado a Documentos JSON?**



2.8. Actividad Interactiva 5

**Prueba tus conocimientos**

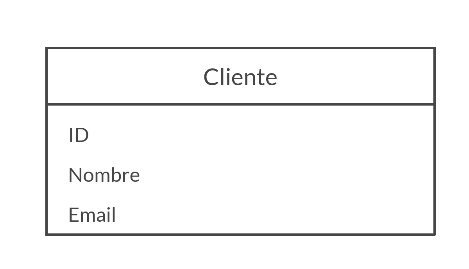
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ es una característica presente en un Sistema de Base de Datos Relacional pero no en uno No Relacional.

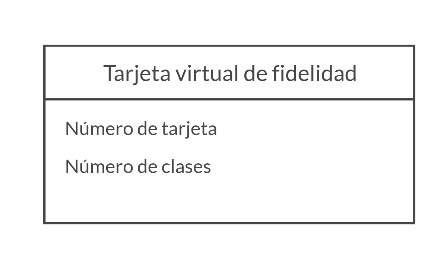


2.9. Ejercicio 2

Desafío práctico

Se está desarrollando un sistema para clientes leales de una Nueva Academia de Zumba. Cada cinco clases el cliente obtiene una gratis. Después de un análisis inicial de los requerimientos del sistema se han identificado las siguientes entidades:





Genera la representación de las entidades, sus atributos y tipos de datos usando los siguientes enfoques de bases de datos:

* Sistema de Bases de Datos Relacional
* Sistema de Bases de Datos No Relacional orientado a Documentos JSON

**Muestra la representación para el cliente cuyo ID es el 1002, cuyo nombre es Ernesto y cuyo email es ernesto@miemail.com y que tiene una tarjeta de fidelidad número 1 y número de clases acumuladas hasta la fecha 3.**

**Solucion:**

- Sistema de Bases de Datos Relacional

|  |  |  |  |
| --- | --- | --- | --- |
| Tabla - Cliente | | |  |
| id | nombre | email | tarjeta\_id |
| 1002 | Ernesto | ernesto@miemail.com | 1 |

|  |  |
| --- | --- |
| Tabla – Tarjeta virtual de Fidelidad | |
| tarjeta\_id | numero\_clases |
| 1 | 3 |

* Sistema de Bases de Datos No Relacional orientado a Documentos JSON

{

“Id”: “1002”,

“Nombre”: “Ernesto”,

Email: “ernesto@miemail.com”

Tarjeta: [{

“Id”: “1”,

“numero\_de\_clases”: 3

}]

}

1. Lección 2: Base de Datos Relacionales
   1. [Definiciones en Base de Datos Relacionales](https://learn.nextu.com/mod/lesson/view.php?id=9659&pid=P_WEB_DATABASE)

Cada entidad se debe transformar en una tabla y cada atributo se debe convertir en campos, los campos son las columnas de las tablas y cada instancia de una entidad es una fila,

Una tabla tiene una única llave primaria que representa una fila, usualmente es una clave de tipo entero.

La clave foránea es un campo que apunta a una clave primaria en otra tabla, para poder establecer relaciones entre tablas. (Asociaciones).

Varios tipos de asociaciones:

1:N – uno a muchos

N:1 – muchos a uno

N-M – de muchos a muchos – Se establece una tabla intermedia con las asociaciones

1:1 – de uno a uno

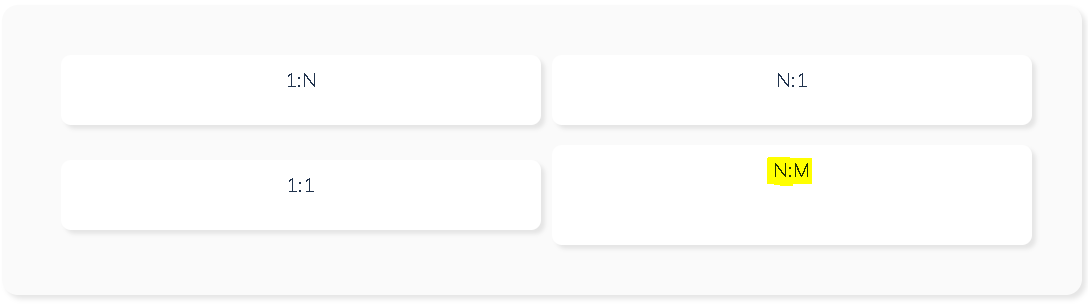
La clave primaria se identifica con una asterisco: ejemplo Id\*

* 1. Actividad Interactiva 1

**Prueba tus conocimientos**

Un nuevo restaurante ha abierto sus puertas en la ciudad. Te invitamos a ayudar a sus dueños a diseñar la base de datos para su nuevo sistema de Especialidades Culinarias. Hasta el momento se han definido dos Entidades: Cocinero y Plato. La tabla cocineros incluye información de los cocineros que laboran en el restaurante, mientras que la tabla platos incluye los platos que este ofrece a sus clientes. Los cocineros del restaurante son seleccionados de forma tal que puedan preparar todos los platos del Menú.

**¿Qué tipo de asociación es la más conveniente entre Cocinero y Plato (en ese orden)?**

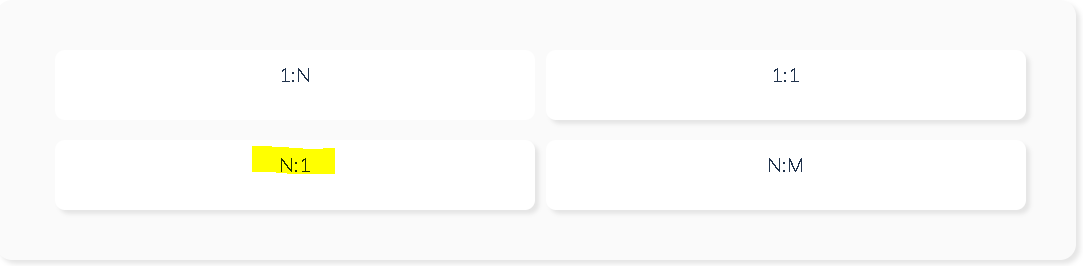


* 1. Actividad Interactiva 2

**Prueba tus conocimientos**

En el restaurante, se ha agregado información acerca de los clientes y se les permite a estos escoger un único plato favorito.

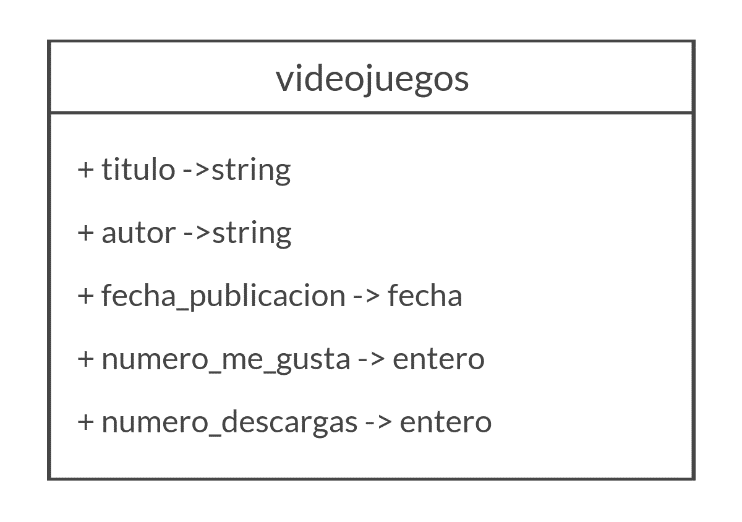
**¿Qué tipo de asociación es la más conveniente entre clientes y platos (en ese orden)?**



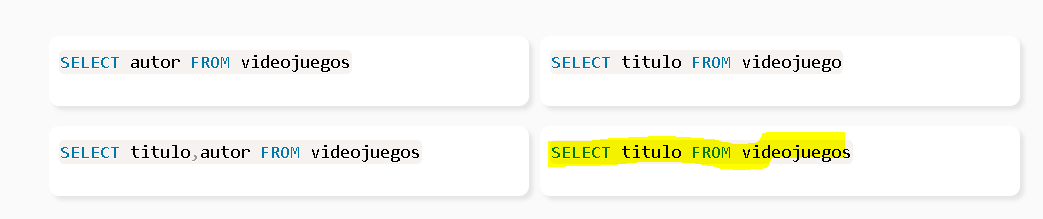
* 1. SQL Sentencia SELECT
  2. Actividad Interactiva 3

Prueba tus conocimientos

MiVideojuego es una tienda que vende juegos de computadoras en línea. Ellos tienen un sistema para el control de las ventas, el cual tiene una base de datos que incluye la tabla videojuegos con todos los videojuegos disponibles en la tienda. Los campos y tipos de la tabla se muestran a continuación:

  
El encargado de la tienda virtual le ha pedido a uno de los miembros del staff que proporcione una lista de todos los títulos de los videojuegos disponibles en la tienda.

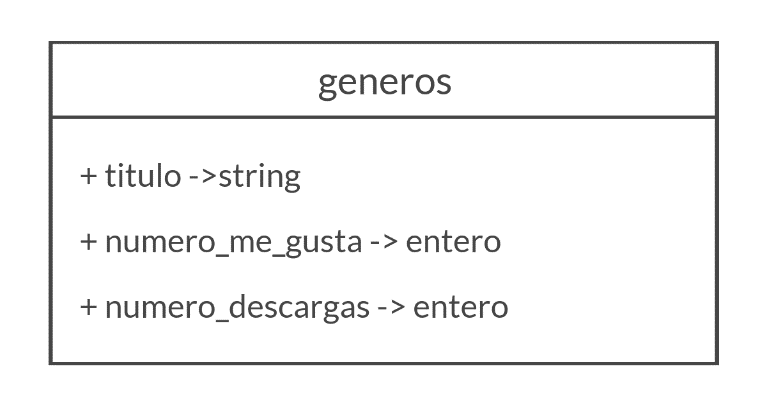
**¿Qué sentencia SQL debería usar el empleado para cumplir con este requerimiento?**



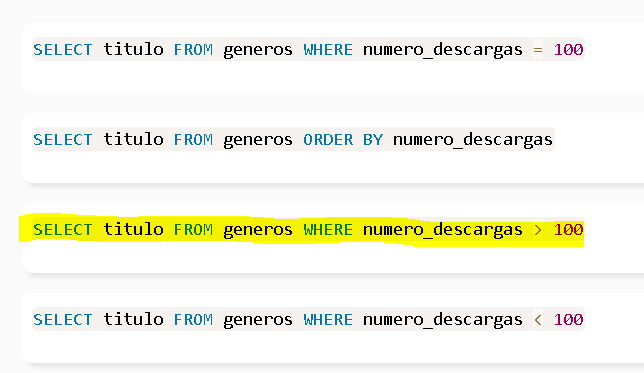
* 1. Actividad Interactiva 4

**Prueba tus conocimientos**

En la tienda MiVideojuego se tiene un sistema para el control de las ventas, el cual incluye la tabla generos la cual contiene todos los géneros de videojuegos disponibles en la tienda. Los campos y tipos de la tabla se muestran a continuación:

  
El encargado de la tienda virtual le ha pedido a uno de los miembros del staff que le proporcione una lista de todos los géneros de los videojuegos disponibles en la tienda que tienen más de 100 descargas

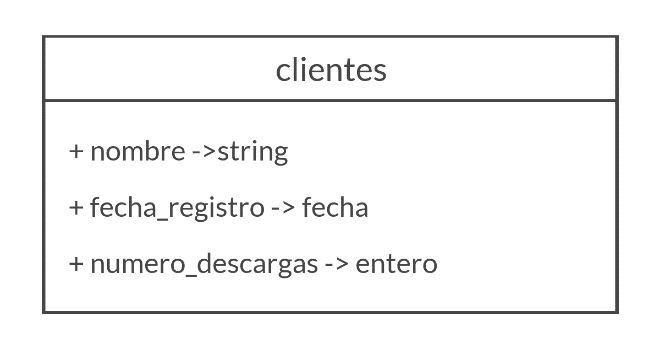
**¿Qué sentencia SQL debería usar el empleado para cumplir con este requerimiento?**



* 1. Actividad Interactiva 5

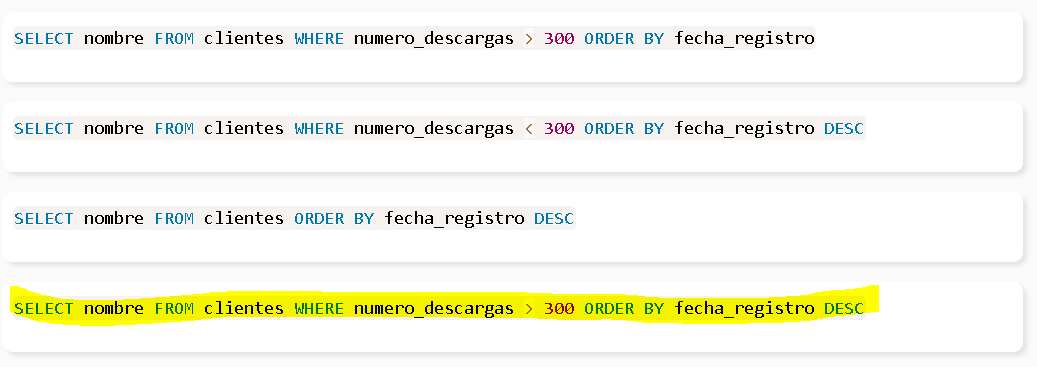
**Prueba tus conocimientos**

En el sistema para el control de ventas de la tienda MiVideojuego, se tiene una tabla con todos los clientes que compran en la tienda. Los campos y tipos de la tabla se muestran a continuación:



El encargado de la tienda virtual le ha pedido a uno de los miembros del staff que le proporcione una lista de todos los nombres de los clientes de la tienda, que tengan un número de descargas mayor a 300, ordenados descendentemente por fecha de registro.

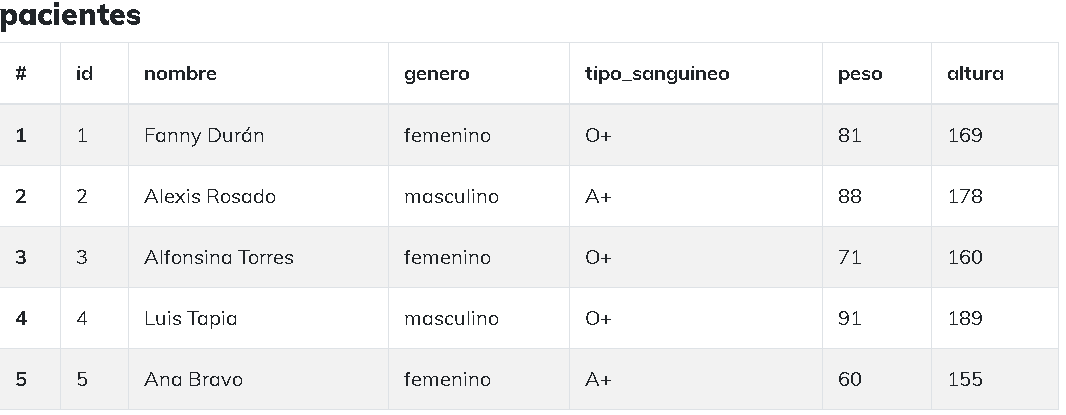
**¿Qué sentencia SQL debería usar el empleado para cumplir con este requerimiento?**



* 1. Ejercicio

**Sandbox**

Dada la tabla paciente con los valores iniciales



Realizar las consultas SELECT para cumplir los siguientes requerimientos:

1. Consultar todos los datos de las pacientes mujeres

Select \* from pacientes where genero like “femenino”

1. Consultar todos los datos de los pacientes mujeres ordenados por altura

Select \* from pacientes where genero like ‘femenino’ order by altura

1. Consultar todos los datos de las pacientes mujeres con tipo sanguíneo O+

Select \* from pacientes where genero like ‘femenino’ and tipo\_sanguineo like ‘O+’

1. Consultar todos los datos de las pacientes mujeres con tipo sanguíneo O+ ordenados por altura descendente

Select \* from pacientes where genero like ‘femenino’ and tipo\_sanguineo like ‘O+’ order by altura desc

1. Consultar el nombre de los pacientes con tipo sanguíneo O+

Select nombre from pacientes where tipo\_sanguineo like ‘O+’

1. Consultar el nombre y la altura de los pacientes hombres con tipo sanguíneo O+

Select nombre,altura from pacientes where genero like ‘masculino’ and tipo\_sanguineo like ‘O+’

1. Consultar el nombre y la altura de los pacientes hombres con tipo sanguíneo O+ y altura mayor que 180

Select nombre, altura from pacientes where genero like ‘masculino’ and tipo\_sanguineo like ‘O+’ and altura > 180

3.9. ¿Sabías qué?

**Breve historia de SQL**



El lenguaje de consulta estructurado recibe su nombre de las siglas en inglés SQL (Structured Query Language).



Las personas no suelen pronunciar SQL como las letras que lo conforman, sino que lo pronuncian como el precursor del mismo que se llama SEQUEL.



SQL fue originalmente creado por la empresa IBM como un sublenguaje de acceso a datos.



Sus primeras implementaciones comienzan a mediados de los años 70.



En 1987 se transformó en estándar ISO de la American National Standards Institute (ANSI).



Ha tenido varios problemas legales de nombre desde su implementación hasta que se definió como SQL.



Aunque en teoría es estándar, cada manejador de base de datos lo extiende con pequeñas variantes.



Ha tenido muchas versiones desde entonces, siendo las más usadas SQL 2000 y SQL 2008.

1. Lección 3: Más SQL
   1. [SQL Sentencias INSERT y DELETE](https://learn.nextu.com/mod/lesson/view.php?id=9660&pid=P_WEB_DATABASE)

INSERT INTO [tabla] VALUES (,) o

INSERT INTO [tabla] ([campo1, campo2]) VALUES (,)

DELETE FROM [tabla] WHERE [condiciones]

* 1. Ejercicio Práctico 1

**Sandbox**

Realizar las sentencias para cumplir los siguientes requerimientos:

* 1. Insertar el registro con id: 14, perteneciente a "Mario Tapia", masculino, con tipo sanguíneo O+, peso 91 y altura 189

INSERT INTO pacientes VALUES (14,"Mario Tapia","masculino","O+",91,155)

* 1. Consultar la tabla actual con la sentencia SELECT \* FROM pacientes

SELECT \* from pacientes

* 1. Insertar el registro con id: 15, perteneciente a "Ana Tapia", femenino, con tipo sanguíneo A+, peso 60 y altura 155

INSERT INTO pacientes VALUES (15,"Ana Tapia","femenino","A+",60,155)

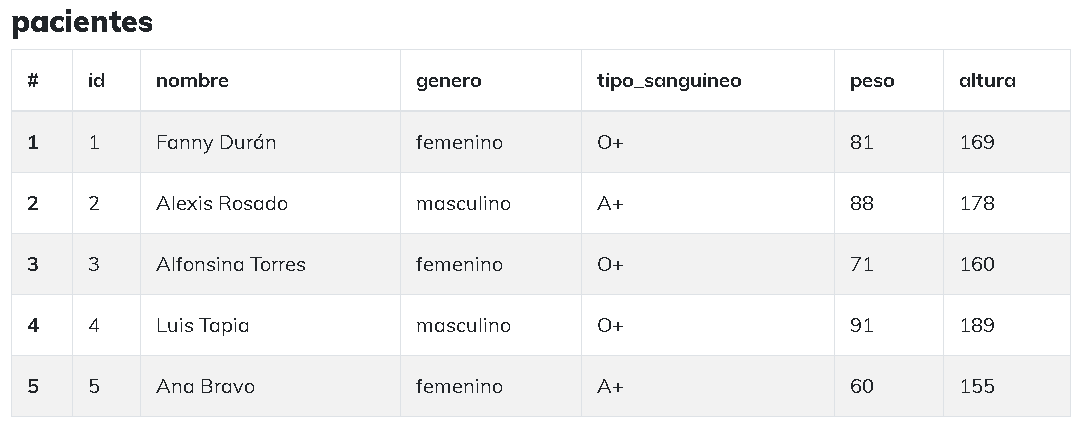
* 1. Consultar la tabla actual con la sentencia SELECT \* FROM pacientes

SELECT \* from pacientes

* 1. Ejercicio Práctico 2

**Sandbox**

Dada la tabla pacientes con los valores iniciales.



Realizar las sentencias para cumplir los siguientes requerimientos:

Tip: Para referirse a una fila en particular se suele usar el valor del campo `id` como condición

1. Eliminar al paciente Fanny Durán

DELETE FROM pacientes WHERE id=1

1. Consultar la tabla actual con la sentencia SELECT \* FROM pacientes para ver los cambios

SELECT \* FROM pacientes

1. Eliminar a todos los pacientes con tipo sanguíneo O+

DELETE FROM pacientes WHERE tipo\_sanguineo = "O+"

1. Consultar la tabla actual con la sentencia SELECT \* FROM pacientes para ver los cambios

SELECT \* FROM pacientes

1. Eliminar a todos los pacientes

DELETE FROM pacientes

1. Consultar la tabla actual con la sentencia SELECT \* FROM pacientes para ver los cambios

SELECT \* FROM pacientes

* 1. Sentencia UPDATE

2 formas de utilizar la sentencia update

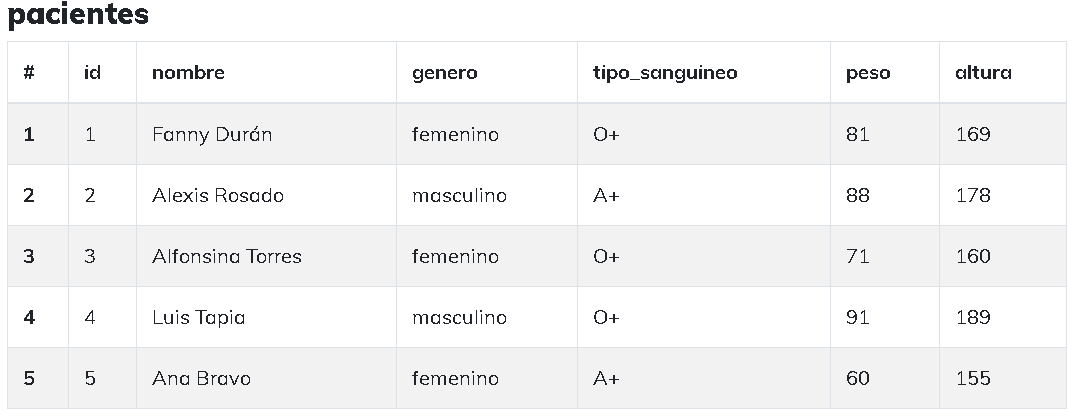
UPDATE [tabla] SET [campo=valor,]

UPDATE [tabla] SET [campo=valor,] WHERE condiciones

* 1. Ejercicio Práctico 3

**Sandbox**

Dada la tabla pacientes con los valores iniciales.



Realizar las sentencias para cumplir los siguientes requerimientos:

Tip: Para refererse a una fila en particular se suele usar el valor del campo `id` como condición

1. Actualizar el peso al Fanny Duran a 72

UPDATE pacientes SET peso=72 WHERE nombre ="Fanny Durán"

1. Consultar la tabla actual con la sentencia SELECT \* FROM pacientes para ver los cambios

SELECT \* from pacientes

1. Actualizar la altura y el peso al Luis Tapia Duran a 180, 85 respectivamente

UPDATE pacientes SET altura=180, peso=85 WHERE nombre="Luis Tapia"

1. Consultar la tabla actual con la sentencia SELECT \* FROM pacientes para ver los cambios

SELECT \* from pacientes

1. Actualizar el tipo sanguíneo a todos los pacientes a O+

UPDATE pacientes SET tipo\_sanguineo="O+"

1. Consultar la tabla actual con la sentencia SELECT \* FROM pacientes para ver los cambios

SELECT \* from pacientes

* 1. ¿Sabías qué?

**Resumen de Comandos SQL**

En esta lectura conseguirás rápidamente los comandos iniciales para trabajar con SQL en sus operaciones básicas (SELECT, INSERT, DELETE, UPDATE).

Recuerda que las palabras encerradas en corchetes [ ] son nombres de tus elementos: tablas, columnas, etc.

**SELECT (Para consultar tablas):**

SELECT [campos] FROM [tabla]

SELECT [campos] FROM [tabla] WHERE [condiciones]

SELECT [campos] FROM [tabla] WHERE [condiciones] ORDER BY [campos y orden(DESC / ASC)]

**INSERT (Para insertar en una tabla):**

INSERT INTO [tabla] VALUES (,) INSERT INTO [tabla] ([campo1], [campo2]) VALUES (,)

**DELETE (Para borrar filas):**

DELETE FROM [tabla] DELETE FROM [tabla] WHERE [condiciones]

**UPDATE (Para actualizar):**

UPDATE [table] SET [campo] = [valor],[campo] = [valor]...

UPDATE [table] SET [campo = valor] WHERE [condiciones]

Esperamos que esta ayuda sea de utilidad para ti en el futuro. Recuerda que estás no son las únicas formas de utilizar esos comandos, sin embargo, son las formas más utilizadas. Conoceremos más variantes de estos comandos próximamente.

1. Lección 4: Bases de Datos No relacionales
   1. [Bases de Datos No Relacionales JSON](https://learn.nextu.com/mod/lesson/view.php?id=9661&pid=P_WEB_DATABASE)
   2. Actividad Interactiva 1
   3. Actividad Interactiva 2
   4. ¿Sabías qué?
   5. Actividad Interactiva 3
   6. Actividad Interactiva 4
   7. Ejercicio
   8. ¿Sabías qué?

6. UNIDAD 2: Prueba