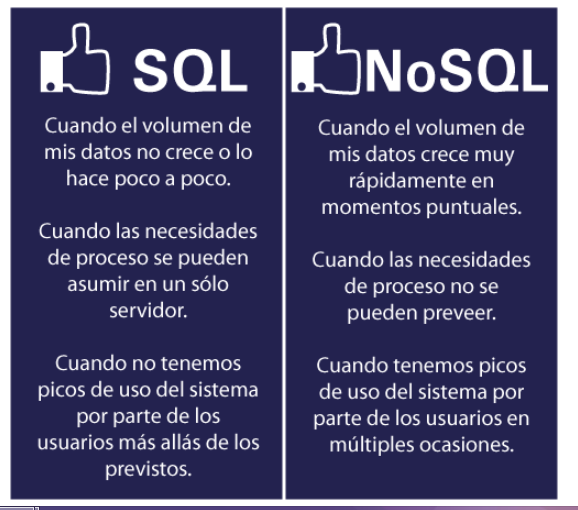
| ***2º ASIR: ASGBD*** | **TAREA 01*:* TEMA 1 Generalidades SGBD**  **BBDD SQL y NoSQL** | *Curso 2022-23*  *1ª Ev* |
| --- | --- | --- |
| *Nombre:* ***Marco Batista Calado*** | |

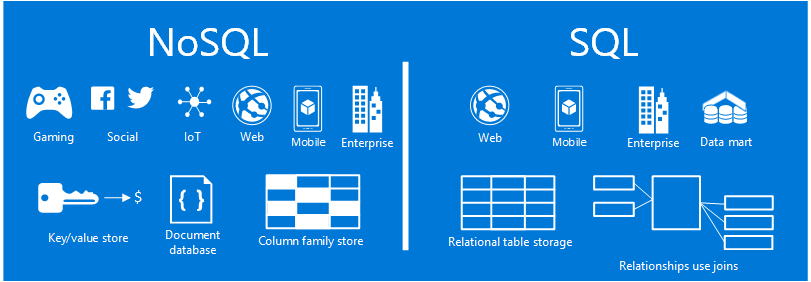
1. A partir del punto 8 de los apuntes en el que hay algo de información sobre bases de datos relacionales y las que no lo son investiga para dejar claro en qué consisten unas y otras y exponlo a continuación:

Imaginemos por un momento una ciudad como Medellín (Colombia), donde todas las personas hablan el mismo idioma, por tanto es la única forma de que todos los residentes se comuniquen e interactúen. **Si se cambia ese idioma se perjudican todos los residentes.**

Las bases de datos relacionales utilizan un lenguaje de consulta estructurado para la manipulación de datos, estas se conforman por filas, columnas y registros y se almacenan por tablas. Para manipular los datos en SQL, se requiere primero determinar la estructura de estos, **si se cambia la estructura de uno de los datos, puede perjudicar todo el sistema, ya que las tablas están relacionadas.**

Ahora imaginemos otra ciudad como Bogotá (Colombia) y pensemos hipotéticamente que en cada uno de los hogares se habla un idioma diferente, todos interactúan distinto y no hay entendimiento entre todos, pero nadie afecta a nadie. Las bases de datos no relacionales tienen un esquema dinámico, no se requiere la estructura de los datos para su manipulación. Los datos se pueden almacenar de cualquier manera, columnas, documentos, gráficos, etc, y cada documento puede tener su propia estructura, sin afectar los demás, puede agregar más campos a medida que se avanza. Están conformadas por documentos, campos y datos del documento, además, se almacenan por colecciones.





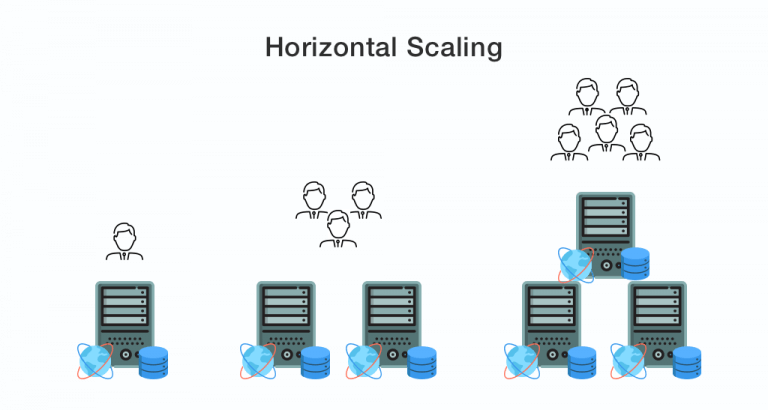
1. En las imágenes del punto 8 aparecen los siguientes términos:

* **NoSQL:** son bases de datos diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Son ampliamente reconocidas porque son fáciles de desarrollar, tanto en funcionalidad como en rendimiento a escala. **Usan una variedad de modelos de datos, que incluyen documentos, gráficos, clave-valor, en-memoria y búsqueda.**

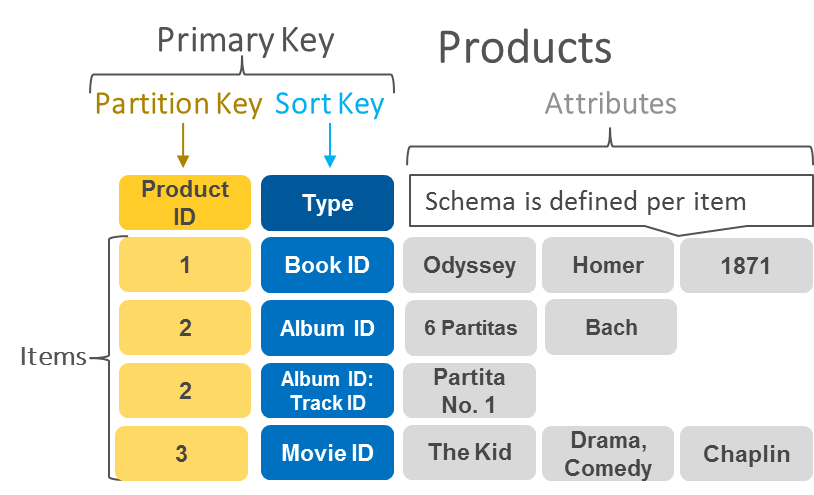
Son las que, a diferencia de las relacionales, no tienen un identificador que sirva de relación entre un conjunto de datos y otros. Como veremos, la información se organiza normalmente mediante documentos y es muy útil cuando no tenemos un esquema exacto de lo que se va a almacenar.

Con relación a formatos, la información de una base de datos puede ser almacenada en tablas o documentos. Cuando los datos son organizados en un archivo de Excel, es en formato tabla, pero cuando simplemente son datos escritos como cartas, fórmulas o recetas, son datos en formato documento. Esto aplica para los dos tipos de bases de datos.

* **Atomicidad:** requiere que cada transacción sea "todo o nada": si una parte de la transacción falla, todas las operaciones de la transacción fallan, y por lo tanto la base de datos no sufre cambios. Un sistema atómico tiene que garantizar la atomicidad en cualquier operación y situación, incluyendo fallas de alimentación eléctrica, errores y caídas del sistema.
* **Consistencia:** se asegura que cualquier transacción llevará a la base de datos de un estado válido a otro estado válido. Cualquier dato que se escriba en la base de datos tiene que ser válido de acuerdo a todas las reglas definidas, incluyendo (pero no limitado a) los constraints, los cascades, los triggers, y cualquier combinación de estos.
* **Aislamiento:** se asegura que la ejecución concurrente de las transacciones resulte en un estado del sistema que se obtendría si estas transacciones fueran ejecutadas una detrás de otra. Cada transacción debe ejecutarse en aislamiento total; por ejemplo, si T1 y T2 se ejecutan concurrentemente, luego cada una debe mantenerse independiente de la otra.
* **Durabilidad:** significa que una vez que se confirmó una transacción (commit), quedará persistida, incluso ante eventos como pérdida de alimentación eléctrica, errores y caídas del sistema. Por ejemplo, en las bases de datos relacionales, una vez que se ejecuta un grupo de sentencias SQL, los resultados tienen que almacenarse inmediatamente (incluso si la base de datos se cae inmediatamente luego).

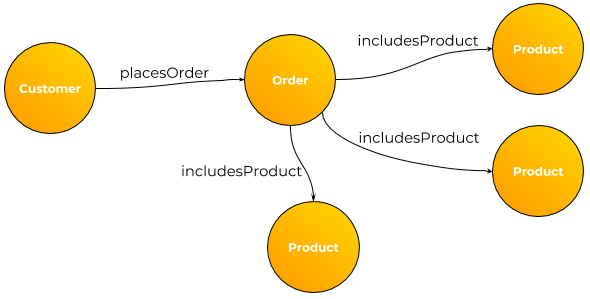


* **Escalar horizontalmente:** implica tener diversos servidores que conocemos como nodos, trabajando como un todo. Se configuran en forma de una red de servidores denominada clúster. Esta tiene la finalidad de dividir eficientemente la demanda de trabajo entre todos los nodos que conforman la red de servidores. Es uno de los principales atributos de las bases de datos NoSQL. Esto debido a que el crecimiento es sumamente extenso y técnicamente no tiene límites. Podríamos agregar cuantos servidores sean necesarios.



* **Clave-valor:** Una base de datos clave-valor almacena datos como un conjunto de pares clave-valor en los que una clave sirve como un identificador único. Tanto las claves como los valores pueden ser cualquier cosa, desde objetos simples hasta objetos compuestos complejos. Las bases de datos clave-valor son altamente divisibles y permiten el escalado horizontal a escalas que otros tipos de bases de datos no pueden alcanzar.



* **Big Table:** es de Google no sql, Bigtable es un mapa distribuido ordenado y multidimensional con tres dimensiones: filas, columnas y marca temporal. ¿Cómo se relacionan entre sí? Es un sistema que divide los datos en columnas para almacenar toda la información en tablas compuestas por celdas. Cada una de esas celdas dispone de una marca temporal que permite visualizar la evolución de ese dato a lo largo del tiempo.
* **BD documentales:** almacenan datos semiestructurados sin un esquema predefinido. La información se almacena en documentos que incluyen todas las características del registro, y las consultas se realizan en base a estos documentos. Esto las diferencia de las bases relacionales que organizan la información en tablas y las consultas se realizan en base a los campos (columnas) de dichas tablas. Las bases de datos documentales se usan principalmente para almacenar grandes cantidades de datos, o cuando se trata de información poco estructurada.
* **BD orientadas a grafos:** No SQL. Tiene su principal característica en que representan la información en vértices y aristas cumpliendo con la disciplina matemática de la Teoría de Grafos. Un grafo estará compuesto por dos elementos: los nodos (vértices) y las relaciones (aristas). Un nodo representa una entidad, en el que almacenaremos piezas de datos o atributos de tipo clave-valor, mientras que las relaciones representan cómo se conectan y se asocian dos nodos.

Explica en qué consiste cada uno de ellos teniendo en cuenta que están relacionados con BD

1. Busca un ranking con las 10 bases de datos más utilizadas en el mes de junio de este año y rellena la siguiente tabla con información sobre ellas.

|  | **Nombre** | **Relacional o NoSql** | **Página oficial** | **Principales características** | **Libre o Comercial** | **Logo** |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Oracle** | **Relacional** | [**Oracle**](https://www.oracle.com/es/) | * Herramienta de **administración gráfica** intuitiva y cómoda de utilizar. * **Control de acceso**. * **Protección de datos** * **Lenguaje de diseño de bases de datos muy completo (PL/SQL)** * **Alta disponibilidad** * **Gestión de usuarios:** agilidad en los trámites, reducción de costes y seguridad en el control de las personas que acceden a las aplicaciones y a los sistemas. * **Multiplataforma:** puede ejecutarse desde un PC hasta una supercomputadora. * Permite el uso de **particiones** para hacer consultas, informes, análisis de datos, etc. * **Soporta todas las funciones** que se esperan de un buen servidor. * Software del servidor que puede ejecutarse en **multitud de sistemas operativos:** Linux, Mac, Windows, etc. | **Comercial** |  |
|  | **MySQL** | **Relacional** | [**MySQL**](https://www.mysql.com/) | * **Arquitectura Cliente y Servidor** * **Compatibilidad con SQL** * **Vistas** * **Procedimientos almacenados:**posee la característica d eno procesar las tablas directamente sino que a través de procedimientos almacenados es posible incrementar la eficacia de nuestra implementación. * **Desencadenantes: p**ermite además poder automatizar ciertas tareas dentro de nuestra base de datos. En el momento que se produce un evento otro es lanzado para actualizar registros o optimizar su funcionalidad. * **Transacciones**   Es una opción razonable para ser usado en el ámbito empresarial. Al estar basado en código abierto permite a pequeñas empresas y desarrolladores disponer de una solución fiable y estandarizada para sus aplicaciones. Por ejemplo, si se cuenta con un listado de clientes, una tienda online con un catálogo de productos o incluso una gran selección de contenidos multimedia disponible, MySQL ayuda a gestionarlo todo debida y ordenadamente. | **Por una parte codigo abierta y cuenta con una versión comercial** |  |
|  | **Microsoft SQL Server** | **Relacional** | [**Descargas de SQL Server | Microsoft**](https://www.microsoft.com/es-es/sql-server/sql-server-downloads) | * **Soporte de transacciones.** * Escalabilidad, estabilidad y seguridad. * Soporte de procedimientos almacenados. * **Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.** * Permite trabajar en modo **cliente-servidor**, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información. * **Permite administrar información de otros servidores de datos.** | **Comercial** |  |
|  | **PostgreSQL** | **Relacional** | [**PostgreSQL**](https://www.postgresql.org/) | * **Alta concurrencia.** * **Soporte para múltiples tipos de datos de manera nativa.** * **Soporte a triggers** * **Trabajo con vistas.** * **Soporte para bases de datos distribuidas.** * **Soporte para gran cantidad de lenguajes.** | **Libre** |  |
|  | **MongoDB** | **NoSQL** | [**MongoDB**](https://www.mongodb.com/) | Base de datos orientada a documentos. Esto quiere decir que en lugar de guardar los datos en registros, guarda los datos en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON.  Esto representa una de las diferencias más importantes con respecto a las bases de datos relacionales. Y resulta que no es que no es necesario seguir un esquema. Los documentos de una misma colección - concepto similar a una tabla de una base de datos relacional -, pueden tener esquemas diferentes.   * **Consultas ad hoc.** * **Indexación.** * **Replicación.** * **Balanceo de carga.** * **Almacenamiento de archivos.** * **Ejecución de JavaScript del lado del servidor** | **Libre** |  |
|  | **Redis** | **No SQL** | [**Redis**](https://redis.io/) | * **Increíblemente rápido**. Los datos se encuentran en la memoria principal, a diferencia de otros sistemas que se encuentran en SSD o disco duro convencional. Como se elimina el acceso a esos discos, se evitan retrasos: se accede y se opera mucho más rápido. * **Estructuras de datos en memoria.** Con Redis podemos guardar claves de diferentes tipos de datos y disponemos de un tamaño máximo de 512MB para una cadena, ya sea texto o datos binarios. También admite listas de cadenas, conjuntos de cadenas sin orden, conjuntos clasificados por puntuación, hashes e HyperLogLogs. Es decir, podemos almacenar en la memoria prácticamente cualquier tipo de dato. * **Replicación y persistencia.** * **Compatibilidad con casi cualquier lenguaje.**   Además, es muy fácil de utilizar, gracias a las diferentes herramientas que aceleran el desarrollo, como:   * **Pub/Sub,** fantástico para sistemas de mensajería instantánea y que permite publicar mensajes en canales para entregar a los suscriptores. * **Claves TTL,** que se destruyen a sí mismas con el tiempo que le indiquemos y que son ideales para evitar datos innecesarios en nuestra base. * **Contadores atómicos,** que nos aseguran que no se creen resultados incoherentes. * **Lua,** un lenguaje de scripts liviano pero poderoso. | **Libre** |  |
|  | **IBM Db2** | **Relacional** | [**IBM Db2 – Software de gestión de datos - España**](https://www.ibm.com/es-es/db2) | * **Integridad** * **Múltiples usos:** Provee la capacidad de hacer frente a múltiples necesidades, desde Procesamiento Transaccional de Misión Crítica (OLTP), hasta análisis exhaustivo de los datos para el soporte a la toma de decisiones (OLAP). * **Escalabilidad:** le permiten almacenar información en un amplio rango de equipos, desde un PC portátil hasta un complejo ambiente de mainframes procesando en paralelo. * **Web enabled para e-business:** Incluye tecnología basada en Web que permite generar aplicaciones en las Intranets y responder a las oportunidades de negocios disponibles en Internet. * **Universalidad:** la única base de datos realmente universal; es multiplataforma (16 plataformas - de las cuales 10 no son de IBM), brinda soporte a un amplio rango de clientes, soporta el acceso de los datos desde Internet y permite almacenar todo tipo de datos: * Texto, Audio, Imágenes y Video (AIV Extender) * Documentos XML ( XML Extender) | **Comercial** |  |
|  | **Microsoft Access** | **Relacional** | [**https://www.microsoft.com/es-es/microsoft-365/access**](https://www.microsoft.com/es-es/microsoft-365/access) | * **Ideal para usuarios individuales y equipos pequeños** * **Más fácil de entender y utilizar que la base de datos cliente-servidor** * **Importación y exportación a otras aplicaciones de Microsoft Office y otras aplicaciones** * **Plantillas listas para que los usuarios habituales creen y publiquen datos** * **Permite construir y publicar bases de datos en la Web sin esfuerzo** * **Una función fácil de usar ‘Tell Me’ para asistencia** * **Permite a los desarrolladores crear soluciones personalizadas utilizando el código VBA** * **Opción de ocultar/mostrar la cinta** * **Permite a los usuarios ver los informes Elimina los informes adicionales** * **Permite la salida de informes en formato PDF** | **Comercial** |  |
|  | **Elasticsearch** | **NoSQL** | [**¿Qué es Elasticsearch? | Elastic**](https://www.elastic.co/es/what-is/elasticsearch) | Es un motor de búsqueda que se basa en Lucene el cual nos permite realizar búsquedas por una gran cantidad de datos de un texto específico. Está escrito en [Java](http://www.manualweb.net/java/) y se basa sobre una licencia Apache.   * **Acceso por API** * **Escalabilidad** * **Orientado a documentos** * **Gran velocidad** | **Libre** |  |
|  | **SQLite** | **Relacional** | [**SQLite**](https://www.sqlite.org/index.html) | Permite almacenar información en dispositivos empotrados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser una PDA o un teléfono celular. SQLite implementa el estándar SQL92 y también agrega extensiones que facilitan su uso en cualquier ambiente de desarrollo. Esto permite que SQLite soporte desde las consultas más básicas hasta las más complejas del lenguaje SQL, y lo más importante es que se puede usar tanto en dispositivos móviles como en sistemas de escritorio, sin necesidad de realizar procesos complejos de importación y exportación de datos, ya que existe compatibilidad al 100% entre las diversas plataformas disponibles, haciendo que la portabilidad entre dispositivos y plataformas sea transparente.   * La base de datos completa se encuentra en un solo archivo. * Puede funcionar enteramente en memoria, lo que la hace muy rápida. * Tiene un footprint menor a 230KB. * Es totalmente autocontenida (sin dependencias externas). * Cuenta con librerías de acceso para muchos lenguajes de programación. * Soporta texto en formato UTF-8 y UTF-16, así como datos numéricos de 64 bits. * Soporta funciones SQL definidas por el usuario (UDF). * El código fuente es de dominio público y se encuentra muy bien documentado. | **Libre** |  |