AWS RDS es un servicio completamente gestionado que facilita la creación, el funcionamiento y el escalado de bases de datos relacionales en la nube de AWS. Ofrece compatibilidad con varios motores de bases de datos, incluidos MySQL, PostgreSQL, SQL Server, Oracle y Amazon Aurora.

En bases de datos relacionales, la información se organiza en tablas que consisten en filas y columnas. Cada tabla representa una entidad o relación, y las relaciones entre las tablas se establecen mediante claves primarias y claves foráneas.

Para implementar el modelo relacional en AWS, primero se selecciona el motor de base de datos adecuado según los requisitos del proyecto. Luego, se crea una instancia de base de datos utilizando AWS RDS y se configura según las necesidades de almacenamiento, rendimiento y seguridad. Se diseñan y se crean las tablas necesarias para representar los datos de manera relacional, estableciendo las claves primarias y foráneas según el modelo de datos.

El Modelo Relacional en AWS Data Modeling AWS RDS se integra fácilmente con otros servicios de AWS para ampliar su funcionalidad, como AWS Lambda para ejecutar código sin servidor, AWS Glue para el procesamiento de datos y ETL, Amazon S3 para el almacenamiento de datos a escala y Amazon Redshift para análisis de datos.

AWS RDS ofrece características de seguridad como la encriptación de datos en reposo y en tránsito, control de acceso basado en roles (IAM), y grupos de seguridad para restringir el acceso a la instancia de base de datos.

#### Diferencia entre base de datos relacionales y no relacionales

Relacionales: Utilizan un modelo de datos tabular, donde la información se organiza en filas y columnas en tablas. Cada fila representa una entidad y cada columna representa un atributo.

Relacionales: Los datos se normalizan para minimizar la redundancia y mantener la integridad de los datos. Las relaciones entre las tablas se establecen mediante claves primarias y foráneas.

Relacionales: Tradicionalmente, las bases de datos relacionales han enfrentado desafíos para escalar horizontalmente debido a su naturaleza tabular y sus relaciones predefinidas.

Relacionales: Suelen priorizar la consistencia ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad), garantizando transacciones completas y confiables. No relacionales: Pueden tener diversas estructuras de datos, como documentos, columnas, grafos o pares clave-valor. No están limitadas por el esquema fijo de tablas como en las bases de datos relacionales.

No relacionales. Los datos pueden ser almacenados de manera desnormalizada, permitiendo redundancia y flexibilidad en la estructura. No tienen la misma rigidez en la definición de relaciones que las bases de datos relacionales.

No relacionales: Suelen ser más escalables horizontalmente, ya que están diseñadas para manejar grandes volúmenes de datos distribuidos en múltiples servidores.

No relacionales: Pueden priorizar la disponibilidad y la partición de red (teorema CAP), sacrificando la consistencia en algunos casos para permitir operaciones distribuldas rápidas.

MySQL. Es un sistema de gestión de bases de datos relacional de código abierto. Ampliamente utilizado en aplicaciones web y empresariales debido a su rendimiento, escalabilidad y comunidad de soporte activa. Admite una variedad de plataformas y es compatible con múltiples lenguajes de programación.

Microsoft SQL Server. Desarrollado por Microsoft, es un sistema de gestión de bases de datos relacional con una amplia adopción en entornos empresariales. Ofrece integración con otras tecnologías de Microsoft, como el framework NET, y proporciona herramientas avanzadas para administración, seguridad y análisis de datos.

Mejores SGBD mas usados

PostgreSQL. Es otro sistema de gestión de bases de datos relacional de código abierto conocido por su robustez, funcionalidades avanzadas y capacidad de escalabilidad. Es especialmente popular en aplicaciones que requieren características como integridad referencial, transacciones ACID y soporte para tipos de datos avanzados.

MongoDB. Es un sistema de gestión de bases de datos NoSQL de código abierto que utiliza un modelo de documentos JSON fiexible para almacenar datos. Ampliamente utilizado en aplicaciones web y móviles, así como en entornos de Big Data, debido a su capacidad de escalabilidad horizontal y su capacidad para manejar datos semi-estructurados.

sistemas de gestión de bases de datos más establecidos y potentes utilizado en grandes empresas y organizaciones. Conocido por su rendimiento, escalabilidad y capacidades empresariales, incluida la gestión de datos espaciales, la gestión de grandes volúmenes de datos y la alta disponibilidad.

# Comparativas de los SGBD mas

MySQL vs PostgreSQL Ambos son sistemas de gestión de bases de datos relacionales de código abierto con sólidas comunidades de usuarios y soporte MySQL tiende a ser más popular en aplicaciones web debido a su facilidad de uso y velocidad, mientras que PostgreSQL es conocido por su conformidad con los estándares ANSI SQL y su amplia gama de características avanzadas, como soporte para JSQN, datos espaciales y operaciones de texto completo.

PostgreSQL vs Oracle.
PostgreSQL es un SGBD de
código abierto con características
avanzadas que compilen
directamente con Oracle en
muchos aspectos. Oracle
Database es un SGBD empresarial
establecido y ampliamente utilizado
en grandes organizaciones para
aplicaciones críticas para el
negocio. Oracle ofrece
capacidades avanzadas de
escalabilidad, alta disponibilidad,
seguridad y gestión de datos que
son especialmente relevantes para
entremos empresantales complejos.

MySQL vs SQL Server. MySQL es una opción popular para aplicaciones web de pequeña a mediana escala y startups, mientras que SQL Server es más comúnmente utilizado en empresas y organizaciones que ya utilizan tecnologías de Microsoft. SQL Server ofrece una integración profunda con otros productos de Microsoft, como el framework. NET y Microsoft Azure, lo que lo hace ideal para entornos que dependen de tecnologías de Microsoft. MongoDB vs DynamoDB.
MongoDB es un sistema de gestión
de bases de datos NoSQL de
código abierto que utiliza un
modelo de datos de documentos
JSON flexible. Es popular en
aplicaciones web y móviles debido
a su capacidad de escalabilidad
horizontal y su facilidad de uso.
DynamoDB es un servicio de base
de datos NoSQL completamente
gestionado por AWS que ofrece
rendimiento de baja latencia y
escalabilidad automática según la
demanda. Es ideal para
aplicaciones en la nube que
requieren alta disponibilidad y
rendimiento a escala.

Creación y Definición de Bases de Datos. Los SGBD permiten crear bases de datos y definir su estructura, incluyendo la creación de tablas, la especificación de tipos de datos, la definición de restricciones de integridad y la configuración de indices.

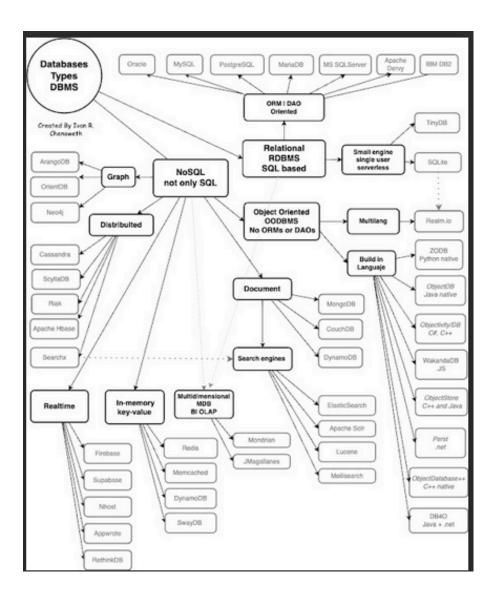
Manipulación de Datos. Los SGBD facilitan la inserción, modificación y eliminación de datos en las bases de datos mediante comandos SQL (Structured Query Language) o interfaces gráficas de usuario. Esto incluye la capacidad de realizar consultas para recuperar información específica de la base de datos.

Gestión de Transacciones. Los SGBD garantizan la integridad y consistencia de los datos mediante el soporte de transacciones, que son secuencias de operaciones que se ejecutan de manera completa y coherente, o no se ejecutan en absoluto. Esto se logra a través del concepto ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).

Funciones de los SGBD

Control de Acceso y Seguridad.
Los SGBD permiten definir y
gestionar permisos de acceso a la
base de datos y a sus objetos
(tablas, vistas, procedimientos
almacenados, etc.), asegurando
que solo los usuanos autorizados
puedan acceder y manipular los
datos.

Optimización del Rendimiento. Los SGBD proporcionan herramientas y técnicas para optimizar el rendimiento de las consultas y operaciones de base de datos, como la creación de índices, la optimización de consultas, la partición de datos y el ajuste de parámetros de configuración.



ES un concepto fundamental en el diseño de bases de datos relacionales que garantiza la consistencia y la validez de los datos almacenados en una tabla. Se refiere a la condición en la que cada fila en una tabla tiene una identificación única y que no puede ser nula, asegurando que cada entidad representada esté correctamente definida y no se dupliquen registros.

Clave Primaria. La integridad de entidad se logra mediante la definición de una clave primaria en una tabla, que consiste en uno o más campos que identifican de forma única cada fila en la tabla. La clave primaria garantiza que no haya filas duplicadas en la tabla y que cada entidad tenga una identificación única.

#### Integridad de Entidad

Restricción de Clave Primaria. En la mayoría de los sistemas de gestión de bases de datos relacionales, se pueden definir restricciones de clave primaria en una tabla para garantizar que no se puedan insertar filas con valores nulos o duplicados en la clave primaria. Esto asegura que la integridad de entidad se mantenga en todo momento y que la tabla cumpla con las reglas de normalización. Beneficios. Garantiza la consistencia y la calidad de los datos almacenados en la base de datos. Facilita la identificación y manipulación de entidades individuales de manera precisa y eficiente. Permite establecer relaciones entre tablas utilizando claves foráneas que hacen referencia a la clave primaria de otras tablas, lo que facilita la creación de relaciones entre

Es un principio en las bases de datos relacionales que garantiza la consistencia y la validez de las relaciones entre tablas. Se asegura de que las relaciones entre las entidades representadas en diferentes tablas sean coherentes y que no se produzcan referencias a datos inexistentes o no válidos.

Claves Foráneas. La integridad referencial se basa en el uso de claves foráneas (o claves externas), que son campos en una tabla que hacen referencia a la clave primaria de otra tabla. Las claves foráneas establecen relaciones entre tablas, permitiendo que los datos de una tabla estén relacionados con los de otra

Restricciones de Integridad
Referencial Los sistemas de
gestión de bases de datos
permiten definir restricciones de
integridad referencial para
garantizar que las relaciones entre
tablas cumplan ciertas reglas. Por
ejemplo, una restricción puede
especificar que no se puede
eliminar un registro de una tabla
principal si hay registros
relacionados en una tabla
secundaria.

Integridad Referencial

Mantenimiento de la Integridad.
Cuando se utiliza integridad
referencial, se garantiza que
cualquier valor en una columna de
clave foránea coincida con un
valor existente en la columna de
clave primaria correspondiente en
otra tabla. Esto evita la creación de
referencias a datos inexistentes y
mantiene la coherencia de las
relaciones entre tablas.

Beneficios. Garantiza la consistencia de los datos al evitar referencias a datos inexistentes. Facilita el mantenimiento de la integridad de la base de datos al automatizar la gestión de relaciones entre tablas. Ayuda a preservar la coherencia y la validez de los datos almacenados, lo que resulta en una base de datos más confiable y fácil de mantener.

Acciones en Cascada. Las restricciones de integridad referencial pueden definirse con acciones en cascada, que especifican qué sucede cuando se actualizan o eliminan registros relacionados. Por ejemplo, se puede configurar una acción en cascada para que, cuando se elimine un registro de la tabla principal, se eliminen automáticamente los registros relacionados en la tabla secundaria.

## Normalizacion en base de datos

Es un proceso de diseño que se utiliza para organizar la estructura de una base de datos relacional de manera eficiente y sin redundancias. El objetivo principal de la normalización es reducir la redundancia de datos y minimizar la posibilidad de anomalías de actualización, inserción y eliminación.

Primera Forma Normal (1FN).
Cada tabla en la base de datos
debe tener una estructura tabular,
lo que significa que cada celda
debe contener un solo valor
atómico y no debe haber valores
repetidos en una misma columna.
Se eliminan los grupos repetitivos
de datos al dividirios en tablas
separadas y relacionarlos a través
de claves primarias y foráneas.

Segunda Forma Normal (2FN).
Una tabla está en segunda forma
normal si cumple con la 1FN y si
cada columna no clave depende
completamente de la clave
primaria. Se elimina la
dependencia parcial de los
atributos no clave dividiendo la
tabla en varias tablas relacionadas.

Tercera Forma Normal (3FN). Una tabla está en tercera forma normal si cumple con la 2FN y si no hay dependencias transitivas entre los atributos no clave. Se eliminan las dependencias transitivas dividiendo las tablas en aquellas que contienen solo atributos que están funcionalmente dependientes de la clave primaria.

Atomicidad (Atomicity). La propiedad de atomicidad garantiza que una transacción se realice completamente o no se realice absoluto. Es decir, una transacción se ejecuta como una unidad atómica e indivisible. Si una parte de la transacción falla, todas las partes de la transacción se deshace un "rollback"), asegurando que la base de datos vuelva a su estado original.

Consistencia (Consistency). La propiedad de consistencia asegura que solo se realicen transacciones que lleven la base de datos de un estado válido a otro estado válido. Esto significa que las restricciones de integridad y las reglas de la base de datos deben mantenerse en todo momento. Si una transacción viola las restricciones de integridad o las reglas de la base de datos, se considera inválida y no se permite que se complete.

Aislamiento (Isolation). La propiedad de aislamiento asegura que el efecto de una transacción sea independiente y no se vea afectado por la concurrencia con otras transacciones que se están ejecutando simultáneamente. Cada transacción debe ejecutarse como si fuera la única transacción en el sistema, evitando interferencias entre transacciones concurrentes.

ACID

Durabilidad (Durability). La propiedad de durabilidad garantiza que los cambios realizados por una transacción que se ha completado con éxito se mantengan permanentemente en la base de datos, incluso en caso de fallo del sistema, apagado repentino o reinicio del servidor. Una vez que se confirma una transacción, sus efectos deben ser permanentes y no reversibles.

## BASE

Básicamente Disponible (Basically Available). Este principio implica que el sistema debe estar siempre disponible para responder a las solicitudes, incluso en situaciones de fallas o particiones en la red. Prioriza la disponibilidad de los servicios para los usuarios, incluso a expensas de la consistencia inmediata de los datos.

Estado Suave (Soft state). El estado suave se refiere a la idea de que el estado del sistema puede cambiar con el tiempo a medida que los datos se replican y se distribuyen entre los nodos. A diferencia de ACID, que requiere que el sistema mantenga un estado consistente en todo momento, BASE permite que el estado del sistema sea momentáneamente inconsistente o "suave" mientras se propagan los cambios.

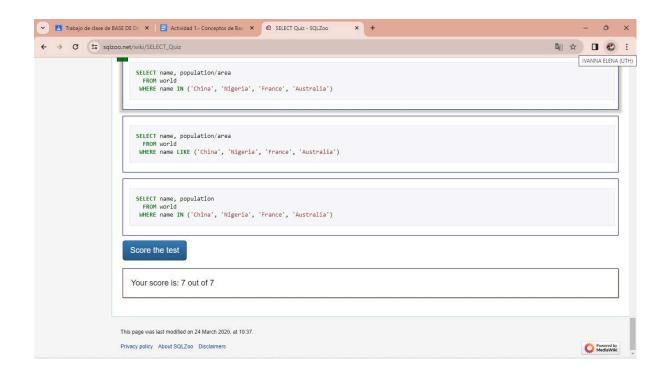
Eventualmente Consistente (Eventually Consistent). La eventual consistencia significa que, aunque el sistema puede experimentar breves períodos de inconsistencia, eventualmente los datos convergerán a un estado consistente en todos los nodos. A medida que las actualizaciones se propagan y se aplican en diferentes partes del sistema, la consistencia se logra eventualmente sin requerir un control estricto en tiempo real.

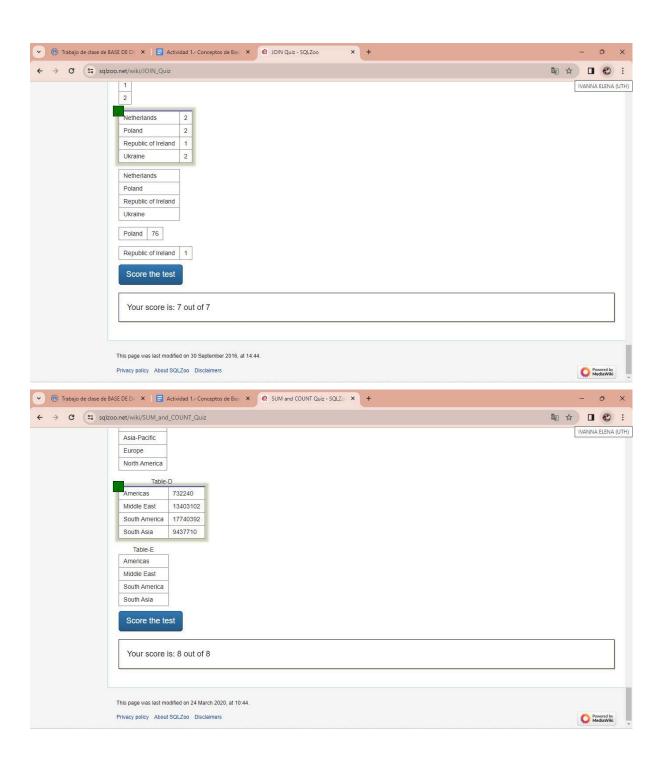
## Teorema CAP

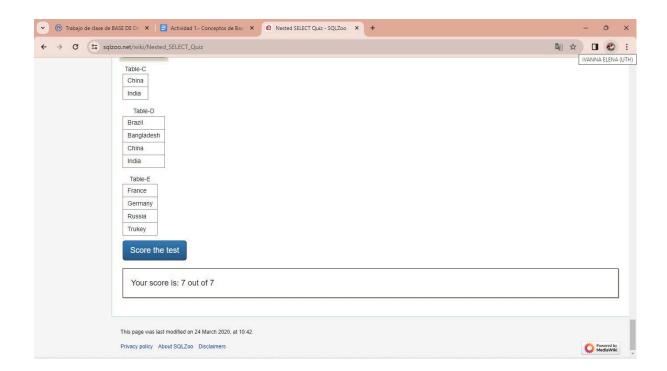
Consistencia (Consistency). Todos los nodos del sistema ven los mismos datos al mismo tiempo, independientemente del nodo al que se acceda. Garantiza que las operaciones de lectura siempre devuelvan la última escritura realizada en el sistema.

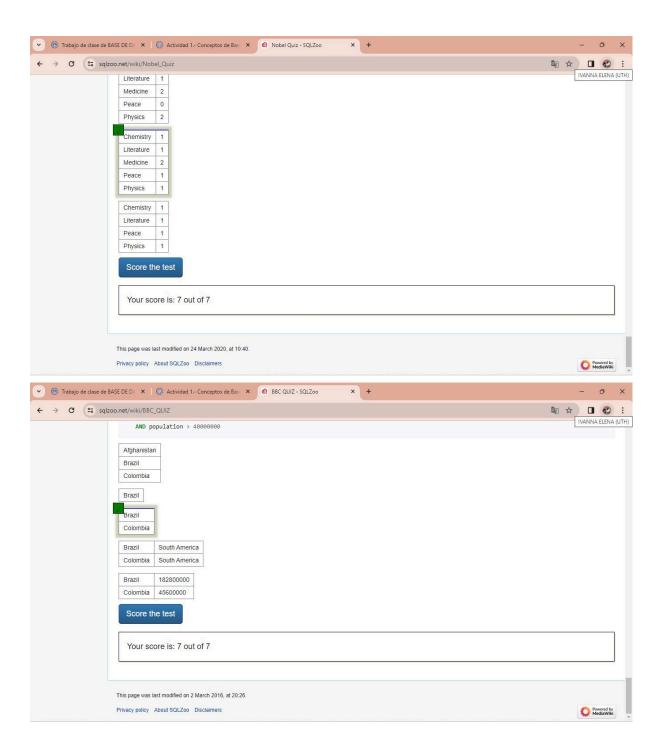
Tolerancia a particiones (Partition Tolerance). El sistema continúa funcionando incluso si la red se divide en múltiples segmentos que no pueden comunicarse entre sí. Permite que el sistema mantenga la disponibilidad y la consistencia a pesar de las fallas de red que pueden provocar la pérdida temporal de comunicación entre nodos.

Disponibilidad (Availability). Cada solicitud recibirá una respuesta, ya sea con éxito o con un error, sin importar el estado actual del sistema. Implica que el sistema sigue funcionando y sirviendo solicitudes incluso en presencia de fallos y particiones en la red.









## Conclusiones

Las bases de datos No Relacionales, ya sean orientadas a objetos, documentales u otros modelos, ofrecen soluciones versátiles para casos de uso específicos, proporcionando flexibilidad y escalabilidad en el manejo de datos en entornos modernos y dinámicos.

Los quizes fueron divertidos y una manera sencilla de recordar y aprender nuevos comandos para las bases de datos.

Bibliografía

CodigoCompilado [@CodigoCompilado]. (2015, agosto 6). Base de datos #13 | Normalización (1FN, 2FN y 3FN). Youtube.

https://www.youtube.com/watch?v=bO18omSzeR4

Diferencias entre base de datos relacional y no relacional. (s/f). SlideShare.

Recuperado

el 31 de enero de 2024, de

https://es.slideshare.net/rafq007/diferencias-entre-base-de-datos-relacional-y-no-relacional

Florencio, A. (2017, mayo 24). Comparativa de los principales sistemas gestores de Bases de Datos (SGBD). Cursos GIS | TYC GIS Formación.

https://www.cursosgis.com/comparativa-de-los-principales-sistemas-gestores-de-bases-de-datos-sgbd/

INTEGRIDAD DE ENTIDAD E INTEGRIDAD REFERENCIAL EN SQL SERVER Y ACCESS. (s/f). SlideShare. Recuperado el 31 de enero de 2024, de

https://es.slideshare.net/vyezk007/integridad-de-entidad-e-integridad-referencial-en-sql-server-y-access

Introduction to Dart. (s/f). Dart.dev. Recuperado el 26 de enero de 2024, de https://dart.dev/language

Jorge Sánchez. Manual de Gestión de Bases de Datos. Diseño lógico de bases de datos

relacionales. (s/f). Jorgesanchez.net. Recuperado el 31 de enero de 2024, de https://jorgesanchez.net/manuales/gbd/diseno-logico-relacional.html

Marín, R. (2019, abril 16). Los gestores de bases de datos (SGBD) más usados. Canal

Informática y TICS.

https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/

NoSQL: Introducción a las Bases de Datos no estructuradas. (s/f). SlideShare.

Recuperado el 31 de enero de 2024, de

https://es.slideshare.net/dipina/nosql-introduccin-a-las-bases-de-datos-no-estruct uradas

SELECT basics. (s/f). Sqlzoo.net. Recuperado el 31 de enero de 2024, de https://sqlzoo.net/wiki/SELECT\_basics

Tipos y función de los gestores de bases de datos. (s/f). Powerdata.es. Recuperado el

31 de enero de 2024, de

http://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/406547/tipos-y-funci-n-de-los-gestores-de-bases-de-datos

(S/f-a). Google.com. Recuperado el 31 de enero de 2024, de

https://drive.google.com/file/d/1cd1v\_06CMIIVBpBBrZ9XNMOoZlead5iZ/view?us p=drive web&authuser=1

(S/f-b). Amazon.com. Recuperado el 31 de enero de 2024, de

https://aws.amazon.com/compare/the-difference-between-acid-and-base-databas