

Preguntas equipo

1. ¿Qué otros SDBD existen actualmente en el mercado?

Actualmente existe una gran variedad de SDBD además de PostgreSQL, algunos ejemplos populares son:

- MySQL
- SQL Server
- SQLite
- MariaDB
- DuckDB
- Oracle
- Db2
- Apache Ignite
- Apache Drill
- Apache Hive

2. ¿Cuáles son las principales diferencias con *PostgreSQL*?

PostgreSQL es una plataforma de aplicación, es decir, un servidor en el contexto de la nube para diseñar, construir y mantener aplicaciones para negocios y organizaciones, en otras palabras, permite la agilidad y colaboración a lo largo del desarrollo de una aplicación.

Una de las cosas que hace especial a *PostgreSQL* es que permite escribir funciones en distintos lenguajes de programación, que se da gracias a su flexibilidad. Por mencionar algunos tenemos PL/Python, PL/Perl, PL/Java y PL/R. Esto nos brinda la posibilidad de usar el lenguaje que más nos convenga para resolver problemas específicos, tal como R para estadística, python para servicios o llamadas a APIs, etc. Todo esto dentro de una sentencia SQL.

Además de lo mencionado, también podemos crear funciones en C y llamarlas. De igual forma, se pueden integrar funciones escritas en lenguajes de programación varios para después llamarlas en una sola consulta SQL.

A diferencia de otros gestores de bases de datos como *MySQL* y *SQL Server*, no se necesita compilar nada para poder crear funciones de agregación (funciones donde los valores de varias tuplas son agrupadas para facilitar algún fin).

Otra diferencia de PostgreSQL con otros SDBD es la relativa facilidad para la creación de nuevos tipos de datos. Dichos tipos de datos pueden ser usados como columns en las tablas. Y no sólo eso, de igual manera se pueden definir nuevos operadores y funciones para facilitar el desarrollo.

PostgreSQL cuenta además con extensiones, las cuales facilitan el uso y descarga de herramientas, como es el caso de los lenguajes de programación antes mencionados.

Algunas otras cosas a recalcar es que es de código abierto, a diferencia de algunos como *MySQL*, *Oracle*, *SQL Server Express*, *Access*, *Db2*, etc.

Para terminar tenemos que *PostgreSQL* permite recuperar información de un punto en el tiempo en caso de desastre. También es tolerante a errores por lo que el sistema no colapsará si algún usuario hace algo indebido. Esto aunado a su disponibilidad en los múltiples sistemas operativos hacen de *PostgreSQL* un buen candidato a tomar en cuenta para las distintas organizaciones.

3. ¿Por qué una empresa debería escoger una base de datos *open source*?

Hay diferentes razones por las que las bases de datos *open source* pueden ser una mejor elección:

En primera, tenemos que al ser *open source* significa que no necesitas pagar una licencia para poder usarla, lo cuál significa menos gastos para la empresa.

Otra ventaja es que como el código *open source* puede verse y modificarse por cualquiera, se puede personalizar y añadir más funcionalidades a la base de datos que se construya con dicho código para adaptarse mejor a las necesidades de la empresa.

También tenemos la facilidad de que si alguna persona nueva quiere trabajar con el código de la base de datos es más probable que al ser *open source* esta nueva persona ya sepa cómo funciona el código y no sea necesario aprender desde cero, lo cuál también le permite a la empresa ahorrarse tiempo en que los nuevos empleados aprendan cómo funciona la base de datos.

Finalmente al tener un código *open source* generalmente existen más recursos y personas que hayan mejorado o implementado diferentes funciones, por lo que es más fácil encontrar soluciones a los posibles errores que surjan al trabajar con la base de datos.

4. ¿Cuáles son las ventajas, para un DBA el trabajar con un *SMBD*, *open source*?

La primera ventaja (y la más evidente, quizá) que ofrece utilizar sistemas manejadores de bases de datos de código abierto, es el costo prácticamente nulo que tienen dichos sistemas, lo que permite que sean instalados en tantos equipos como sea necesario y se evite tener que adquirir y renovar licencias.

Debido a que las comunidades que utilizan y desarrollan estos sistemas de código abierto suelen ser bastante grandes, es sencillo encontrar soporte de manera gratuita, lo que facilita el mantenimiento de estos sistemas y la resolución de problemas.

Nuevamente, al ser tan grandes las comunidades de usuarios y desarrolladores de los *SMBD* de código abierto, es más difícil encontrarse con problemas de compatibilidad.

El libre acceso al código, permite que, en caso de ser necesario, se agreguen o modifiquen funcionalidades al sistema.

Debido a que estos sistemas son probados por miles de personas, es muy poco probable que se encuentre en ellos errores que pongan en riesgo nuestra información.

5. ¿Qué son las bases de datos *NoSQL*? Menciona 3 ventajas y desventajas contra las bases relacionales.

Como bien sabemos, una base de datos relacional basa su estructura en tablas, algunos ejemplos de sistemas manejadores de bases de datos que siguen este modelo podrían ser MySQL y Oracle, Así que podemos inferir que una Base de Datos *NoSQL* es una que no es de tipo relacional o dicho de diferente manera, que no utiliza tablas como otras bases de datos, las bases de datos *NoSQL*

están orientadas a los documentos y le permiten almacenar y recuperar datos en otros formatos. Algunas plataformas NoSQL conocidas son MongoDB y Redis.

Ventajas:

- Las bases de datos NoSQL son más flexibles y escalables al trabajar con tipos de datos complejos y que cambian constantemente.
- Al trabajar con una base de datos NoSQL, es posible puede agregar datos nuevos sin tener que definirlos anteriormente en el esquema de la base de datos.
- Es muy útil cuando no tenemos un esquema exacto de lo que se va a almacenar ya que este no es necesario.

Desventajas:

- Algunas bases de datos NoSQL son incapaces de tener en cuenta la atomicidad de las instrucciones o la integridad de los datos.
- En muchas ocasiones se requiere de un amplio conocimiento de la herramienta para su mantenimiento por circunstancias como puede ser que la administración es usualmente por consola debido a la falta de una interfaz gráfica.
- No existe una estandarización como sí la tienen las bases de datos relacionales. Existe un futuro incierto en cuanto a bases de datos NoSQL se refiere.

Referencias y bibliografía

- Qué es una base de datos NoSQL — Base de datos no relacional de Rackspace. (s. f.). Rackspace Technology. <https://www.rackspace.com/es-mx/library/what-is-a-nosql-database>
- Team, P. F. (2022, 29 septiembre). NoSQL vs SQL: principales diferencias y cuándo elegir cada una de ellas. Pandora FMS - The Monitoring Blog. <https://pandorafms.com/blog/es/nosql-vs-sql-diferencias-y-cuando-elegir-cada-una/>
- Obe, R. O., & Hsu, L. S. (2017). *PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database*. O'Reilly Media.
- Sharma, K. (2021, Enero 19). *What is PostgreSQL and why do enterprise developers and start-ups love it?* Ubuntu. Recuperado en Febrero 26, 2023, de <https://ubuntu.com/blog/what-is-postgresql>