**MySQL vs POSTGRESQL**

**UBER**

En un inicio la empresa internacional UBER quien proporciona a sus clientes una red de transporte privado, a través de su software de aplicación móvil, basó su arquitectura en una aplicación única escrita en lenguaje Python, conectada a una base de datos PostgreSQL. Con el tiempo, esta arquitectura se ha renovado, ahora es planteada en modelos de micro servicios y nuevas plataformas de datos. Entre ellas destaca el cambio de PostgreSQL por Schemaless, una capa de abstracción sobre MySQL que construye una base de datos distribuida. Este cambio se ve originado inicialmente por dificultades presentadas al cabo de un tiempo, quizá gracias a la expansión de negocio que representaba UBER.

En el momento en que UBER eligió esta PostgreSQL como base de datos para su negocio, no imaginó los alcances de crecimiento que podría presentar su negocio, mucho menos imaginaron que su servicio de software fuese tan exitoso. UBER comienza identificar algunas falencias en sus BDs, es su funcionamiento transaccional y demás, como lo es actualización de versiones, lo que le hace percatarse de las diferentes dificultades que esta presentaba, así que la empresa comienza una investigación ardua y realiza algunas comparaciones con otras bases de datos, con otras que pudiesen adaptarse a sus requerimientos, su constante expansión en el mercado, por lo que toma la decisión de migrar su base de datos PostgreSQL a MySql, presentado está ultima beneficios como, acceso de base de datos ligero, velocidad y performance en la mejora y optimización de Consultas.

A continuación se hacen algunas comparaciones, que pudieron determinar el cambio de las bases de datos de UBER a ciencia cierta.

**MySQL**

En este sistema de base de datos tiene una gran ventaja ya que fue creado con el propósito de obtener gran velocidad, el consumo de recursos tanto en CPU como en memoria es menor, además de presentar mayor velocidad al conectarse al servidor, y a la ejecución de consultas, entre otras. También cuenta con mejores utilidades de administración como son, backup, recuperación de errores, etc. también es seguro ya que no suele perder la información ni corromper los datos , cuenta con una muy buena y mejor integración con PHP Además no hay límites en el tamaño de los registros. Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos. MySQL se comporta mejor que POSTGRESQL a la hora de modificar o añadir campos a una tabla "en caliente", aunque MySQL presenta algunas desventajas, como lo es no soportar transacciones, "roll-backs" ni subselects. No considera las claves ajenas, e ignora la integridad referencial.

**POSTGRESQL**

Este sistema de base de datos es un administrador de base de datos que a su vez es capaz de soportar un modelo de datos que consiste en una colección de relaciones con nombre, los cuales suelen contar con atributos específicos. POSTGRESQL intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase, por lo que posee algunas ventajas como lo es que gracias a su arquitectura le va muy bien el aumento de CPUs y así mismo a su memoria RAM, es capaz de soportar claves ajenas con comprobaciones de integridad referencial, cuenta con un buen y mejor rendimiento para soportar trigges y procedimientos en el servidor. Además soporta un subconjunto de SQL92 mayor que el que soporta MySQL. Además, tiene ciertas características orientadas a objetos. Así como este sistema tiene sus ventajas también tiene sus desventajas e inconvenientes como, consume muchos más recursos y carga más el sistema, tiene un límite del tamaño de cada fila de las tablas a 8k (se puede ampliar a 32k recompilando, pero con un coste añadido en el rendimiento). Es de 2 a 3 veces más lenta que MySQL. Menos funciones en PHP.

Haciendo referencia a la importante conclusión de UBER, acerca del inevitable cambio de arquitectura, se cita al ingeniero de software *Evan Klitzke, para Uber (https://eng.uber.com/mysql-migration/):*

*“*Postgres nos sirvió bien en los primeros días de Uber, pero nos topamos con problemas significativos que escalaban a Postgres con nuestro crecimiento. Hoy en día, tenemos algunas instancias heredadas de Postgres, pero la mayoría de nuestras bases de datos están construidas sobre MySQL (normalmente usando nuestra capa [Schemaless](https://eng.uber.com/schemaless-part-one/" \t "_blank) ) o, en algunos casos especializados, con bases de datos NoSQL como Cassandra. En general, estamos muy contentos con MySQL, y es posible que tengamos más artículos de blog en el futuro explicando algunos de sus usos más avanzados en Uber.”