

## Autonomía

En este concepto a lo que se refiere a las bases de datos distribuidas indica que cada instancia de base de datos puede operar de diferente manera e independientemente en función de los factores tales como los componentes del sistema, y además pueden ejecutar transacciones independientes y están permitidos modificarlos.

La conducta en la cual la individualidad de las bases de datos procesa queries y los optimizan no son afectados durante la ejecución global de todos y estos mismos nos permiten tener acceso a diferentes bases de datos y ya sea que la consistencia de los sistemas no debería estar comprometida cuando se le agregan mas bases de datos o dejan de estar en el sistema distribuido.

Por otro lado, las dimensiones de la autonomía pueden ser especificadas como las siguientes:

- Designando la autonomía individual en las bases de datos ya que son libres de usar cualquier modelo de datos y técnicas de transacción o las que ellos prefieran
- La autonomía comunicativa en cada base de datos individual es libre de tomar cualquier decisión en qué tipo de información desea proveer a las otras bases de datos o en cualquier otro software que controla la ejecución global.
- La ejecución autónoma en cada base de datos puede ejecutar las transacciones que sean presentadas y como quieren que sean presentadas

También podemos clasificar las diferentes alternativas por importancia, una de ellas es tight integration donde una simple imagen de la base de datos esta disponible para cualquier usuario para poder compartir información en la cual reside en una sola base de datos

La siguiente clasificación es semiautomática en la cual consiste en que las bases de datos pueden operar independientemente, pero tienen que consultar a una federación para decidir que tipo de información puede ser compartida y además estas deciden que base de datos puede tener acceso a cualquier usuario.

Y la ultima alternativa es es total isolation donde el sistema individual están en stand-alone de las bases de datos en la que no conoce la existencia de las bases de datos adjuntas.

## Distribución

En este punto veremos como se puede distribuir la información, aquí tomaremos en cuenta dos modelos diferentes: cliente-servido y peer to peer. En el modelo cliente-servidor.

En el modelo cliente-servidor concentra las actividades en el servidor mientras que el cliente está enfocado en darle un ambiente de manejo incluyendo una interfaz de usuario para que sea más fácil manipular los datos. La importancia de este modelo es que tenemos que definir quien es el cliente y quien es el servidor para poder distribuir bien los datos de las bases de datos.

El otro modelo para considerar es el peer to peer, aquí no hay distinción entre los servidores y los clientes, todas las máquinas están llenas de bases de datos que se comunican entre sí con otras máquinas para poder ejecutar queries y transacciones y casi la gran mayoría de estos sistemas manejan este tipo de distribución

## Heterogeneidad

La importancia de este punto es que el lenguaje de los queries sea solo uno y que las transacciones tengan sus protocolos de seguridad y representar la heterogeneidad con una herramienta única para el poder de un solo modelo y las limitaciones de las mismas herramientas de estas. La heterogeneidad no involucra meterse en diferentes paradigmas en los diferentes modelos de datos, pero sí cubre los diferentes lenguajes que puede manejar el sistema y que usen el mismo modelo de datos. En SQL está estandarizado para que su uso sea más fácil de manejar y además se maneja un solo modelo de datos.