



Bienvenidos a la segunda presentación dedicada a los modelos de agregación.

# Modelos de agregación

- Características
- Consideraciones de diseño
- Casos de uso

EIMT.UOC.EDU

Esta presentación se divide en tres secciones.

La primera resume las principales características asociadas a todos los modelos de agregación. Recordemos que los modelos de agregación incluyen el modelo clave-valor, el documental y el orientado a columnas.

En la segunda y tercera sección, respectivamente, veremos algunas consideraciones de diseño y algunos casos de uso de los modelos de agregación.

# Modelos de agregación

- Características
- Consideraciones de diseño
- Casos de uso

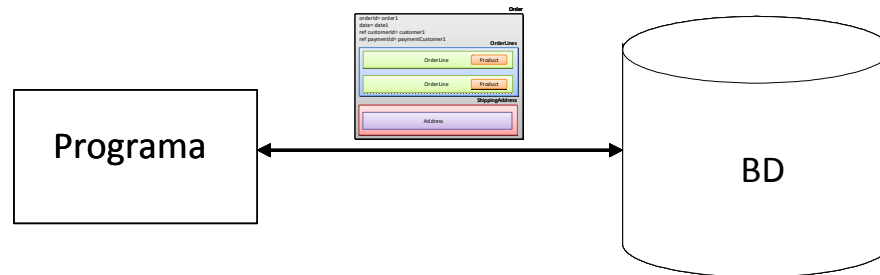
EIMT.UOC.EDU

A continuación vamos a resumir las características más relevantes asociadas a los modelos de agregación, así como sus principales beneficios e inconvenientes.

Recordad que estas cuestiones, en parte, han sido tratadas a través de un ejemplo en la primera presentación dedicada a los modelos de agregación.

## Características

- El agregado es la unidad de acceso y manipulación.
- El agregado puede ayudar a reducir los accesos necesarios para recuperar los datos.
- La consistencia se mantiene a nivel de agregado. Así, por ejemplo, la BD garantiza la definitividad de los cambios producidos a nivel de cada agregado.



EIMT.UOC.EDU

En los modelos de agregación, el agregado constituye la unidad mínima individualmente controlada por el sistema gestor de la base de datos a efectos de acceso y manipulación.

Tal y como muestra el diagrama, el sistema gestor de la base de datos recuperará (o guardará), a petición de cada programa, los agregados necesarios.

Si el diseño de agregados es el adecuado, las operaciones (especialmente las de lectura) serán muy eficientes, dado que se benefician de los efectos de la desnormalización.

Los cambios que el programa efectúe sobre el agregado se harán definitivos de forma atómica en la base de datos. Estos cambios pueden afectar a diferentes partes del agregado.

## Características

- Dependiendo del diseño de agregados efectuado, hay objetos que pueden quedar repartidos en diferentes agregados.
- La responsabilidad del mantenimiento de la consistencia en objetos repartidos en múltiples agregados es de la aplicación.



Los modelos de agregación pueden introducir redundancias en los datos.

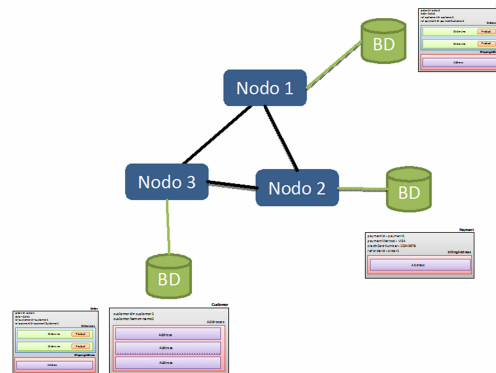
Esto implica que datos referidos a unos mismos objetos del mundo real pueden quedar repetidos en diferentes agregados, como consecuencia de una decisión de diseño. Éste es el caso de las direcciones postales de nuestra base de datos de ejemplo del carrito de la compra, tal y como muestra la figura.

Si ante cambios en esos objetos, el dominio de aplicación requiere que se mantenga la coherencia en el valor de los objetos, garantizar esa coherencia es responsabilidad del programa (y en consecuencia de los personas que intervienen en su creación) y no del sistema gestor de la base de datos.

La situación descrita también es aplicable a referencias entre agregados. Sobre nuestro ejemplo, recordemos que el agregado pedido contiene referencias a otros agregados (en concreto, al pago y al cliente que realiza el pedido). Los programas desarrollados serán los responsables de mantener la validez de dichas referencias.

## Características

- El agregado constituye la unidad a efectos de distribución de los datos, en el caso de que la base de datos sea distribuida.
- Para la distribución de agregados se usan principalmente técnicas de *hash*
- Para aumentar la disponibilidad se usa replicación



EIMT.UOC.EDU

Los modelos de agregación han sido impulsados por corporaciones como Amazon o Google. Esto implica que han sido desarrollados para ser utilizados en entornos altamente distribuidos.

Es por ello que el agregado constituye la unidad ideal para distribuir los datos entre las diferentes bases de datos (o nodos) que conforman el sistema distribuido.

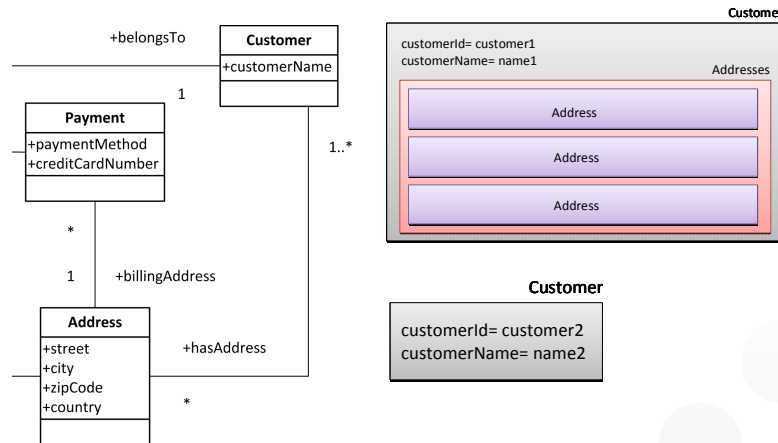
Además, para aumentar la disponibilidad de los datos, los agregados acostumbran a estar replicados, es decir, van a existir diferentes copias de un mismo agregado en diferentes nodos. En la figura, podemos ver que éste es el caso de un agregado de tipo pedido que está almacenado en los nodos 1 y 3.

Las réplicas de un mismo agregado, cuando no son idénticas, constituyen una fuente de inconsistencia. El sistema gestor de la base de datos se responsabilizará de mantener la coherencia de las réplicas existentes de un mismo agregado.

Para distribuir los agregados entre los diferentes nodos, en general, se usan técnicas de *hash* (un ejemplo sería una técnica denominada *consistent hashing*). Estas técnicas garantizan que los agregados (y sus réplicas, si las hubiera) se repartirán de forma uniforme entre todos los nodos, con el objetivo de equilibrar la carga de trabajo. También facilitan la adición de nuevos nodos al sistema, causando molestias mínimas en el resto de nodos que podrán seguir suministrando servicio a los usuarios. Esto se relaciona con la afirmación que las bases de datos NoSQL favorecen los esquemas de crecimiento horizontal (esta noción también se conoce como escalabilidad horizontal).

# Características

- Los modelos de agregación presentan un esquema flexible (*schemaless*).



EIMT.UOC.EDU

Los modelos de agregación presentan un esquema flexible.

Esto quiere decir, en primer lugar, que no es necesario definir de forma explícita el esquema de los datos. El esquema está definido de forma implícita en función de la estructura de los datos insertados. En algunos casos, incluso, el esquema es inexistente desde el punto de vista del sistema de gestor de la base de datos. En estas situaciones, el esquema queda implícito en los programas que acceden a la base de datos.

En segundo lugar, la estructuración de agregados de un mismo tipo puede variar. Éste es el caso que ilustra la figura donde tenemos dos agregados de clientes. El primero representa un cliente para el cual tenemos registradas tres direcciones postales. El segundo representa un cliente para el que no tenemos registrada ninguna dirección. Aunque no se muestra en la figura, destacar que incluso podrían existir agregados de tipo cliente que sólo definan una parte de la estructura asociada a la dirección postal (por ejemplo, que incluyan la calle y ciudad, pero no el código postal y país) o agregados de tipo cliente que definan la estructura en un orden diferente al mostrado en la figura.

## Características

- Para facilitar la extracción de información útil a partir de los agregados recuperados de la base de datos, la aplicación puede usar infraestructuras de procesamiento externo, como sería el caso del *framework* MapReduce.

EIMT.UOC.EDU

Finalmente, para facilitar la extracción de información útil a partir de los agregados recuperados de la base de datos, las aplicaciones pueden usar infraestructuras de procesamiento externo, como sería el caso del *framework* MapReduce.

MapReduce es un modelo de programación paralela que permite dividir una consulta en diferentes subconsultas más simples (llamadas *map*) y una función para integrar los resultados de las mismas (llamada *reduce*). Las funciones *map* se ejecutan de forma paralela en cada nodo que contenga datos a procesar. Esto tiene dos ventajas principales: 1) minimiza el tiempo de respuesta (ya que parte de la consulta se ejecuta en paralelo) y 2) minimiza el transporte de datos irrelevantes a través de la red (ya que los datos son consultados localmente en los nodos donde se almacenan y los datos irrelevantes para la consulta se descartan localmente).

Sobre nuestro caso del carrito de la compra se podría utilizar, por ejemplo, para contar el total de pedidos que se han hecho en una fecha determinada. Para ello, se crearía una función *map* que se ejecutará localmente en cada nodo y que identificará los pedidos que se realizaron en la fecha de interés. Por su parte, la función *reduce* se encargará de recoger y contar esos pedidos.

Es importante destacar que el *framework* MapReduce también puede ser aplicado en bases de datos relacionales.



# Modelos de agregación

- Características
- Consideraciones de diseño
- Casos de uso

Una vez vistas las principales características asociadas a los modelos de agregación, pasamos a comentar brevemente algunas consideraciones de diseño a tener en cuenta a la hora de trabajar con estos modelos.

## Consideraciones de diseño

- El diseño de agregados es de vital importancia y está guiado por las funcionalidades que debe proveer la aplicación.
- Los modelos de agregación son una buena solución cuando:
  - Las funcionalidades que tiene que proveer la aplicación están claramente fijadas a priori y no se esperan que vayan a variar significativamente.
  - Las funcionalidades presentan poco solapamientos en cuanto a necesidades de datos.
  - No existen interrelaciones complejas en el dominio de aplicación que se desea representar.
  - Los datos están sujetos a pocos cambios, especialmente en aquellos datos que aparecen repetidos en diferentes agregados.

EIMT.UOC.EDU

Uno de los elementos más importantes para el uso eficaz de modelos de agregación tiene que ver con el diseño de los agregados.

El diseño de agregados debe estar guiado por las funcionalidades a proveer por las aplicaciones que usen la base de datos.

En general, los modelos de agregación son una buena solución en los siguientes casos:

En primer lugar, cuando las funcionalidades estén claramente fijadas a priori y no se espera que vayan a variar en demasía. Si las funcionalidades cambian de manera drástica o se desconocen, el diseño de agregados efectuado puede resultar ineficaz.

En segundo lugar, cuando las funcionalidades a implementar tienen poco solapamiento en cuanto a necesidades de información. En este caso, será posible minimizar las repeticiones de datos entre agregados de diferente tipo.

En tercer lugar, cuando no existan interrelaciones (o asociaciones) complejas en el dominio de aplicación a representar. En general, éste será el caso de modelos conceptuales donde predominen asociaciones jerárquicas (asociaciones con multiplicidad 1 a \*), asociaciones 1 a 1 y relaciones de tipo parte-todo. Fijémonos que nuestro ejemplo del carrito de la compra verifica este criterio.

Y finalmente, cuando los datos estén sujetos a pocos cambios, es decir, cuando las operaciones que se realicen sobre los agregados sean mayoritariamente de creación (o inserción) de agregados y consulta de agregados.

# Modelos de agregación

- Características
- Consideraciones de diseño
- Casos de uso

Para acabar, vamos a enumerar algunos casos de uso que se ajustan a las características previamente descritas.

## Casos de uso

- Almacenamiento de datos de sesiones de usuarios
- Perfiles de usuario
- Aplicaciones de *e-commerce*
- Gestión de contenidos
- Web *analytics* y analíticas en tiempo real
- Almacenamiento de series temporales

EIMT.UOC.EDU

Estos casos de uso son:

- Sistemas que guardan datos sobre datos de sesiones de usuario.
- Aplicaciones que almacenan perfiles de usuario.
- Aplicaciones de *e-commerce*, como la que hemos presentado en nuestro ejemplo del carrito de la compra.
- Aplicaciones de gestión de contenidos, como sería el caso prensa digital, blogs etc.
- Web analytics y analíticas en tiempo real.
- Almacenamiento de series temporales como sería el caso de flujos de datos proveniente de sensores (*Internet of Things*).

# Modelos de agregación

- Características
- Consideraciones de diseño
- Casos de uso

EIMT.UOC.EDU

Aquí finaliza la segunda presentación dedicada a los modelos de agregación.

En ella hemos examinado las características más relevantes asociadas a los modelos de agregación. Asimismo se han incluido algunas consideraciones de diseño y se han enumerado algunos dominios de aplicación donde el uso de modelos de agregación resulta adecuado.

## Referencias

C. Coronel, S. Morris & P. Rob (2013). *Database Systems: Design, Implementation and Management 10e*. Course Technology, Cengage Learning.

E. Redmond, J Wilson (2012). *Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement*. The Pragmatic Bookshelf.

I. Robinson, J. Webber & E. Eifren (2013). *Graph Databases*. O'Reilly.

P.J. Sadalage & M. Fowler. (2013). *NoSQL Distilled. A brief Guide to the Emerging World of Polyglot Persistence*, Pearson Education. (<http://bit.ly/1koKhBZ>).

EIMT.UOC.EDU

Esperamos que hayáis disfrutado y aprendido con este vídeo. A continuación encontraréis algunas referencias que os permitirán profundizar más en los fundamentos de este modelo.

Que tengáis un buen día.