

Calidad de Datos

- Introducción
- Dimensiones de Calidad
- Modelo de Calidad
- Limpieza de Datos

Introducción

- En general la gente lo reduce a “exactitud de datos” (accuracy), sin embargo, es un concepto multi-facético, donde existen diferentes dimensiones.
- Generalizando, los consumidores quieren que los datos:
 - sean relevantes para su uso
 - sean correctos y sin inconsistencias
 - sean lo más actualizados posible
 - se vean en forma adecuada a sus aplicaciones
 - se accedan fácilmente

Introducción- Problemas de Calidad

- Datos incorrectos
 - Errores de digitación, de formato, etc.
- Datos inconsistentes con la realidad
- Datos inconsistentes entre sí
- Datos desactualizados
- Información incompleta
- Datos poco confiables debido a su fuente
- Otros...

Introducción – consecuencias de mala calidad

- Graves consecuencias en la eficiencia y efectividad de organizaciones y negocios
- Algunos reportes (EEUU)
 - Reporte de *DW Institute*: problemas de calidad de datos le cuestan a los negocios del país, más de 600 billones de dólares por año.
 - Entre 50% y 80% de registros electrónicos de criminales resultaron ser incorrectos, incompletos, o ambiguos.
 - Servicio de Correo: de 100000 unidades de correspondencia masiva enviada, 7000 no llegó a destino debido a direcciones incorrectas
 - Más de 35% of de los proyectos de IT fracasan debido a mala calidad de datos.

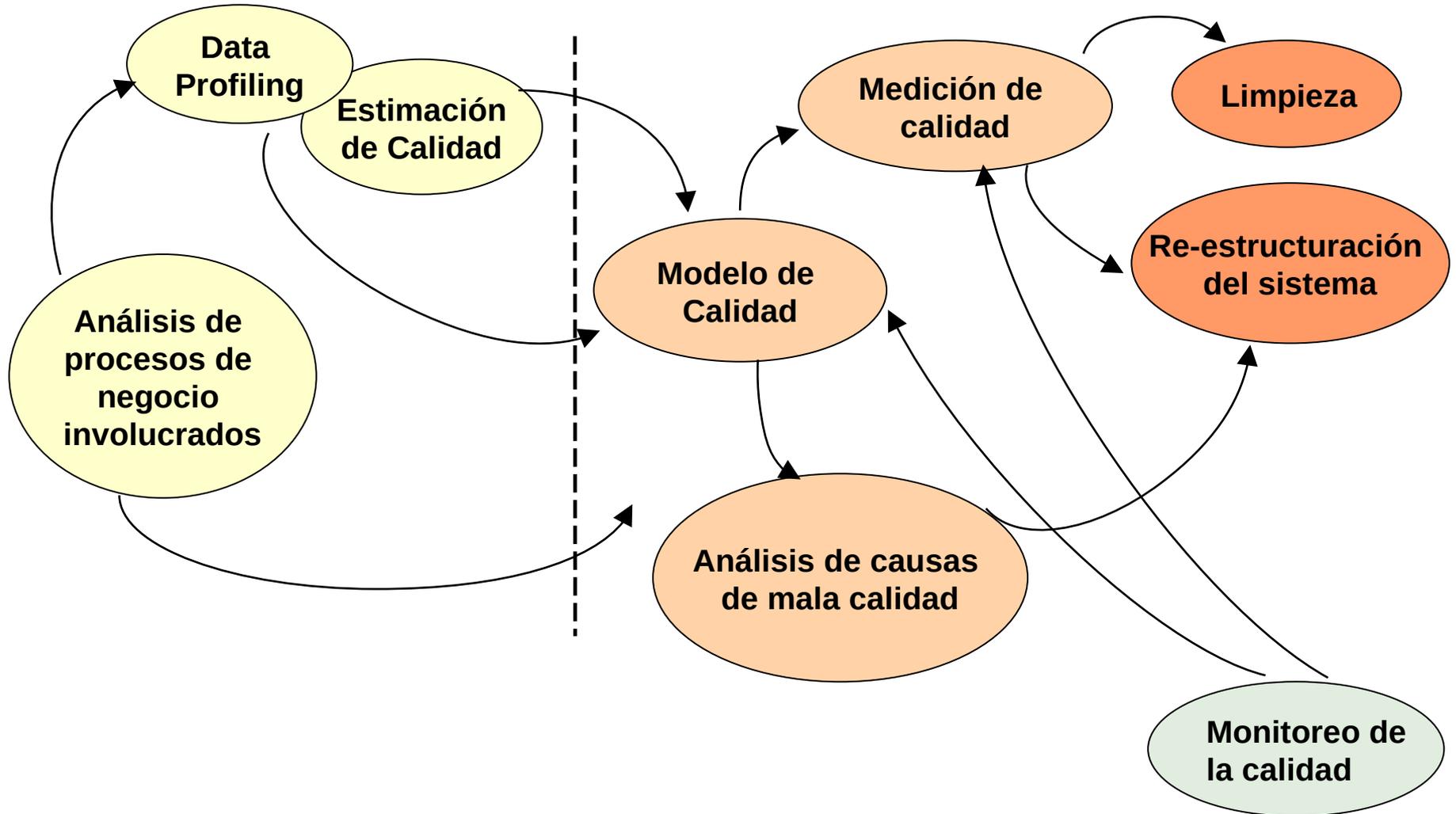
Introducción – consecuencias de mala calidad

- Mala calidad de datos en las empresas...
 - Baja la satisfacción de los clientes
 - Lleva a costos altos e innecesarios
 - Baja la satisfacción en el trabajo y alimenta la desconfianza en la organización
 - Tiene impacto en la toma de decisiones
 - Dificulta la re-ingeniería
- Los avances de la tecnología de la información incrementan el impacto de la mala calidad de datos
 - Datos disponibles a más gente (ej, internet)
 - Gente tiende a suponer que los datos “si están en una computadora están bien”

Introducción – causas de mala calidad

- Producción de los datos
 - Recolección de datos mediante ingreso humano
 - Problemas sistemáticos con la recolección de datos
 - Diferentes fuentes con representaciones diferentes del mismo objeto de la realidad
 - No mantenimiento al día de los datos
- Almacenamiento
 - Formatos diferentes
 - Ausencia de formatos definidos
- Procesamiento
 - Transformaciones incorrectas aplicadas a los datos
 - Integración de datos hecha en forma incorrecta

Gestión de calidad



Dimensiones de Calidad

- La calidad se caracteriza vía múltiples dimensiones o atributos que ayudan a calificar los datos (Multi-dimensionalidad de la calidad).
- Dimensión de calidad:
 - Una dimensión captura una faceta (a alto nivel) de la calidad.
 - Ejemplos:
 - *Frescura*: los datos son recientes/actualizados.
 - *Exactitud*: los datos son exactos/correctos.
 - *Compleitud*: disponemos de todos los datos.

Dimensiones de calidad - Factores

- Factor de calidad:
 - Un factor representa un aspecto particular de una dimensión de calidad.
 - Ejemplo: Varios aspectos de la dimensión *Exactitud* son:
 - *Correctitud semántica*: si los datos representan entidades/estados del mundo real.
 - *Correctitud sintáctica*: si los datos no tienen errores sintácticos.
 - *Precisión*: si los datos tienen el suficiente nivel de detalle.
 - Un factor puede ser más adecuado que otro para algún tipo de problema o aplicación.

- Una dimensión puede verse como un agrupamiento de factores de calidad que tienen el mismo propósito.

Medición de la calidad

- Métrica de calidad: es un instrumento que define la forma de medir un factor de calidad.
 - Se debe definir:
 - La **semántica**
 - Ej. cantidad de valores nulos, cantidad de tuplas, tiempo transcurrido desde la última actualización
 - Las **unidades** de medición.
 - Ej. tiempo de respuesta en ms, volumen en GB, cantidad de transacciones/seg.
 - La **granularidad** de la medida.
 - Ej. cantidad de errores en toda la tabla o en un atributo.
 - Granularidades típicas: celda, tupla, atributo, vista (conj. de celdas), tabla, grupo de tablas, fuente
- Un mismo factor de calidad puede medirse con diferentes métricas.

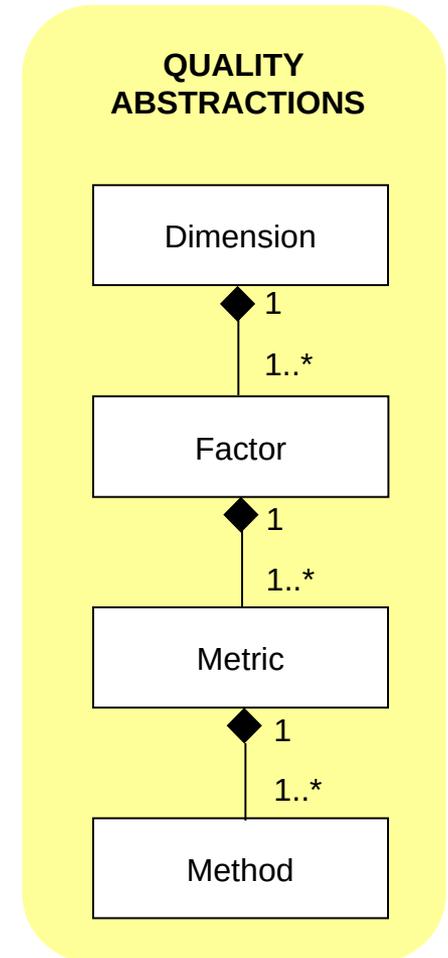
Medición de la calidad

- Método de medición:
 - Un método es un proceso que implementa una métrica.
 - Es el encargado de tomar una serie de medidas (correspondientes a una métrica) para una BD concreta.
 - La implementación del método es dependiente de la aplicación en concreto y de la estructura de la BD
 - Ejemplo: para medir el tiempo transcurrido desde la última actualización, se puede:
 - Usar timestamps de la BD
 - Acceder a los logs de actualización
 - Comparar versiones de la BD
 - ...

- Una misma métrica puede ser medida por diferentes métodos.

Jerarquía de los conceptos de calidad

- En resumen:
 - Las dimensiones representan las facetas de la calidad a alto nivel.
 - Cada dimensión puede refinarse en un conjunto de factores que representan aspectos particulares.
 - Cada factor puede medirse con varias métricas.
 - Cada métrica puede implementarse con varios métodos de medición.



Ejemplo de conceptos de calidad

■ Dimensión:

- **Exactitud**: Concierno la correctitud y la precisión con que los datos del mundo real son representados en un sistema de información

■ Factor:

- **Correctitud sintáctica**: Indica qué tan libre de errores sintácticos están los datos

■ Métricas:

- **Corr. Sint. Booleana**: Un booleano que indica si un dato es sintácticamente correcto o no. (*Ej. un teléfono es correcto o no*)
- **Desviación de corr. sint.**: La distancia a un dato considerado como sintácticamente válido (*Ej. Montevideo, Mtdo*)

■ Métodos:

- **CheckRule**: Chequea si un dato satisface una regla de formato.
- **CheckDictionary**: Chequea si un dato se encuentra en un diccionario.
- **ComputeDistance**: Calcula la distancia entre un dato y el valor más cercano en un diccionario.

Modelo de Calidad

- Para poder tratar la calidad de un SI es necesario definir un modelo de calidad adecuado a las necesidades y prioridades de los consumidores de los datos en ese SI
- Se debe determinar y especificar
 - Dimensiones y factores a medir
 - Métricas y datos o grupos de datos donde éstas se aplican
 - Base de Metadatos de calidad donde se registrarán los resultados de la medición
- Para determinar las dimensiones y factores de calidad existen distintos métodos de trabajo
 - En general, esto incluye la priorización de los datos a medirles la calidad

Modelo de Calidad

■ Métodos

□ Relevamiento **requerimientos de usuarios**

■ 2 opciones:

□ A partir de los **datos prioritarios** defino dimensiones-factores-métricas.

□ Primero defino **dimensiones-factores**, luego selecciono datos, luego defino métricas.

□ A partir de **perfil de los datos** (Data Profiling)

■ Sin intervención de usuarios finales.

■ El perfil de los datos me da las nociones de por dónde están las fallas de calidad. A partir de eso defino las dimensiones-factores-métricas.

□ Orientado al proceso de negocio

Data Profiling

- Primera aproximación al conocimiento sobre los datos del SI que queremos evaluar
 - Su estructura (si los metadatos son consistentes con los datos)
 - Sus relaciones
 - Su volumen
 - Sus problemas y frecuencia de los mismos
 - Patrones que se cumplen
- Algunas técnicas, como
 - Estadísticas básicas
 - Análisis de metadatos
 - Análisis de patrones
 - Detección automática de foreign keys

Data Profiling

- Análisis de atributos solapados de diferentes relaciones
 - Redundancias, claves foráneas
- Valores faltantes o erróneos
 - Cardinalidad actual vs. cardinalidad esperada (cant. clientes)
 - Frecuencia de valores nulos, máximo/mínimo, etc.
- Duplicados
 - Número de tuplas vs. cardinalidad del dominio del atributo
- Claves difusas y dependencias funcionales difusas
 - Restricciones de integridad que no están explícitamente definidas pero que son satisfechas en la mayoría de los casos (un atributo que es clave, dependencias funcionales)

Data Profiling

- “Profiling” con SQL

```
SELECT MIN(A), MAX(A), COUNT(DISTINCT A)
FROM DataTable;
```

```
SELECT City, COUNT(*) AS Cnt
FROM Customer
GROUP BY City ORDER BY Cnt;
```

```
SELECT COUNT(distinct City)
FROM Customer C1, Customer C2
WHERE C1.City = C2.City AND
      C1.Country <> C2.Country;
```

Limpieza de datos

- Identificar y eliminar inconsistencias, discrepancias y errores en datos, para mejorar la calidad
- Otros nombres:
 - “Data cleaning”,
 - “Data cleansing”,
 - “Data scrubbing”

Prevencción de errores

- Localización (o detección) y corrección de errores *no previenen errores futuros*.

- Ej.: Suponer que un proceso crea o reemplaza 1000 registros nuevos o existentes cada día, cada registro tiene 20 campos y la tasa de errores del proceso es 2%. *400 nuevos errores se producen por día. A fin de año se habrán producido 140000 errores.*



Enorme tarea de limpieza.

- Se busca
 - identificar causas (*root-causes*) de los errores
 - eliminar esas causas
 - asegurar que se mantendrá esa ganancia

Resumen

- Dimensiones de Calidad
 - Ordenan los diferentes aspectos de calidad que se pueden medir en los datos.
 - Involucran distintos aspectos, métricas y métodos de medición.
 - Se construye un *Modelo de Calidad*

- Evaluación de la calidad de un Sistema de Información
 - A través de la *medición* de la calidad de los datos.

- Limpieza de datos
 - Tareas dependientes de la dimensión de calidad.
 - Según el caso, distintos grados de automatización

Bibliografía

- Data Quality: Concepts, Methodologies and Techniques. Carlo Batini, Monica Scannapieco. 2006 Springer-Verlag, ISBN-10 3-540-33172-7
- Data Quality for the Information Age. Thomas C. Redman. 1996 Artech House Inc., ISBN 0-89006-883-6
- Information Quality: Fundamentals, Techniques and Use. Felix Naumann, Kai-Uwe Sattler. EDBT Tutorial, Munich, 2006.
- Data Quality. The Accuracy Dimension. Jack E. Olson. Morgan Kaufmann Publishers, Elsevier. 2003. ISBN-10 1-55860-891-5
- B. Otto, K. M. Huner, and H. Osterle, “Identification of Business Oriented Data Quality Metrics,” presented at the ICIQ, 2009, pp. 122–134.

Otros modelos de datos

- Introducción
- Bases de Datos de clave-valor
- Bases de Datos de Documentos
 - XML
 - JSON
- Bases de Datos de Grafos
 - RDF
 - Neo4j
- Resumen

Introducción

- Modelo visto en el curso:
 - Bases de Datos Relacionales (BDR)

- Existen otros modelos de datos:
 - Key-Value stores (clave valor)
 - Bases de datos documentales
 - Bases de grafos

Bases de Datos de clave-valor

- Una base de datos de clave-valor
 - tipo de base de datos NoSQL
 - utiliza un método simple de clave/valor para almacenar datos.
- Almacenamiento de datos diseñado para almacenar, recuperar y gestionar *arrays* (diccionarios).
- Diccionarios: colección de objetos, o registros, que a su vez tienen muchos campos diferentes dentro de ellos, cada uno contiene datos.
- Los registros se almacenan y recuperan utilizando una clave que identifica exclusivamente el registro y se utiliza para encontrar rápidamente los datos dentro de la base de datos.

Bases de Datos de clave-valor

- La clave de un par clave-valor es única.
- En teoría, la clave podría ser cualquier cosa.
 - Depende del DBMS.
 - Algunos DBMS imponen limitaciones y otros no.
- El valor puede ser cualquier cosa:
 - texto (largo o corto),
 - un número,
 - código HTML, código de programación como PHP,
 - una imagen,
 - una lista
 - otro par clave-valor encapsulado en un objeto.
- Algunos DBMS especifican un tipo de datos para el valor. Ejemplo, el valor debe ser un entero.

Bases de Datos de clave-valor

- Ejemplos simples de BD clave-valor:

| Key | Value |
|------|----------------|
| Paul | (123) 456-7890 |
| Bob | (234) 567-8901 |

| Key | Value |
|----------------|------------|
| artist:1:name | AC/DC |
| artist:1:genre | Hard Rock |
| artist:2:name | Slim Dusty |
| artist:2:genre | Country |

- Este ejemplo utiliza una lista como el valor.

| Key | Value |
|-----------|-----------------------|
| 123456789 | APPL, Buy, 100, 84.47 |
| 234567890 | CERN, Sell, 50, 52.78 |

Bases de Datos de Documentos

- Bases de datos NoSQL
- BD NO relacional que almacena datos como documentos estructurados.
 - Normalmente en formatos XML o JSON.
- Las primeras se construyeron en base al estándar de documentos XML (*eXtensible Markup Language*).

BD de Documentos: XML

- El creciente volumen de documentos XML dentro de las organizaciones fue lo que motivó algún tipo de sistema de gestión de documentos XML.
- Una plataforma que implementa los distintos estándares XML, como:
 - XQuery: un lenguaje de consulta para consultar documentos XML
 - XSLT: lenguaje para transformar documentos XML en formatos alternativos, tales como HTML.

BD de Documentos: XML

- Proporciona servicios:
 - almacenamiento,
 - indexación,
 - seguridad
 - acceso concurrente de archivos XML.

- Las BD XML suelen utilizarse como sistemas de gestión de contenido.
 - Organizan y mantienen colecciones de archivos de texto en formato XML (Documentos académicos, de negocios, etc.).

BD de Documentos: XML

- XML se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Ejemplo de documento XML:

```
<item rdf:about="http://www.example.com/item1-info-page.html">
<title>Train tickets to Mountain View</title>
<description>Luxury train tickets</description>
<g:expiration_date>2006-12-20</g:expiration_date>
<g:from_location>San Francisco, CA, USA</g:from_location>
<g:id>1</g:id>
<g:image_link>http://www.example.com/image1.jpg</g:image_link>
<link>http://www.example.com/item1-info-page.html</link>
<g:location>1600 Amphitheatre Pkwy, Mountain View, CA, 94043</g:location>
<g:price>250</g:price>
<g:price_type>starting</g:price_type>
<g:quantity>204</g:quantity>
<g:to_location>Mountain View, CA, USA</g:to_location>
<g:travel_date_range>
  <g:start>2006-12-20T23:00:00</g:start>
  <g:end>2006-12-21T08:30:00</g:end>
</g:travel_date_range>
</item>
```

BD de Documentos: XML

- XML tiene ventajas como formato estándar para documentos. Sin embargo, tiene algunos inconvenientes como formato de almacenamiento.
 - Por ejemplo, es costoso de procesar.
- Las BD XML son el punto de partida para las modernas BD de documentos JSON.
- JSON (*JavaScript Object Notation*) ofrece mayores beneficios y ha logrado mayor popularidad que su predecesor XML.

BD de Documentos: JSON

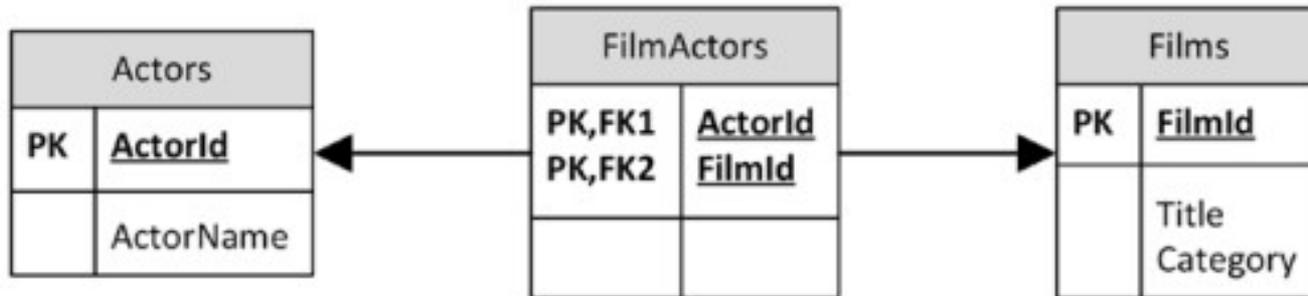
- Es un formato de intercambio de datos.
- Apuntan a soportar aplicaciones web operacionales.
 - almacena y modifica contenido dinámico y los datos transaccionales en el núcleo de las aplicaciones web.
- JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia.
 - Por ejemplo Yahoo y Google.

BD de Documentos: JSON

- La jerarquía de almacenamiento suele ser la siguiente:
 - un documento es la unidad básica de almacenamiento
 - un documento comprende uno o más pares clave-valor (key-value)
 - puede contener documentos anidados y *arrays* (*matrices*).
 - Los *arrays* también pueden contener documentos que permitan una estructura jerárquica compleja.
 - Un *data bucket* o colección es un conjunto de documentos que comparten un propósito común.
 - Los documentos de una colección no tienen que ser del mismo tipo, aunque es común que representen cierta información.

BD de Documentos: JSON

- La siguiente es una BDR de películas y actores:



- En una BD de documentos JSON:

```
{ "_id" : 97, "Title" : "BRIDE INTRIGUE",
  "Category" : "Action",
  "Actors" :
    [ { "actorId" : 65, "Name" : "ANGELA HUDSON" } ]
}
```

```
{ "_id": 115, "Title": "CAMPUS REMEMBER",
  "Category": "Action",
  "Actors" :
    [ Actor document
      [ { "actorId": 8, "Name": "MATTHEW JOHANSSON" },
        { "actorId": 45, "Name": "REESE KILMER" },
        { "actorId": 168, "Name": "WILL WILSON" } ]
    ]
}
```

Annotations: "Film Document" (vertical arrow on the left), "Array of actors" (vertical arrow on the right), "Collection" (vertical arrow on the far right).

```
{ "_id" : 105, "Title" : "BULL SHAWSHANK",
  "Category" : "Action",
  "Actors" :
    [ { "actorId" : 2, "Name" : "NICK WAHLBERG" },
      { "actorId" : 23, "Name" : "SANDRA KILMER" } ]
}
```

BD de Documentos: JSON

- Documentos embebidos : Los actores se anidan como un *array* dentro de los documentos de películas.
- Ejemplo: conjunto de IDs de actores en el documento de películas.



BD de Documentos: XML y JSON

- Las BD de documentos XML y JSON son similares. Sin embargo, cada uno está diseñado para soportar casos de uso bastante diferentes y aplicaciones significativamente diferentes.
- Las BD de documentos se diferencian de otros sistemas relacionales y no relacionales por su adopción de un formato de documento (XML o JSON), como modelo de datos.

Comparando XML y JSON

```
films.json
1 {
2   "Category": "Documentary",
3   "Description": "A Epic Drama of
4   "Length": "86",
5   "Rating": "PG",
6   "Rental Duration": "6",
7   "Replacement Cost": "20.99",
8   "Special Features": "Deleted Sce
9   "Title": "ACADEMY DINOSAUR",
10  "_id": 1,
11  "Actors":
12  [ {
13    "First name": "PENELOPE",
14    "Last name": "GUINNESS",
15    "actorId": 1
16  }, {
17    "First name": "CHRISTIAN
18    "Last name": "GABLE",
19    "actorId": 10
20  }, {
21    "First name": "LUCILLE",
22    "Last name": "TRACY",
23    "actorId": 20
24  }, {
25    "First name": "SANDRA",
26    "Last name": "PECK",
27    "actorId": 30
28  }, {
29    "First name": "JOHNNY",
30    "Last name": "CLOE"
31  }
32 ]
33 }

films.xml
1 <?xml version="1.0" encoding="UTF-8" ?>
2   <Category>Documentary</Category>
3   <Description>A Epic Drama of a Femi
4   <Length>86</Length>
5   <Rating>PG</Rating>
6   <Rental Duration>6</Rental Duration
7   <Replacement Cost>20.99</Replacemen
8   <Special Features>Deleted Scenes,Be
9   <Title>ACADEMY DINOSAUR</Title>
10  <_id>1</_id>
11  <Actors>
12    <First name>PENELOPE</First nam
13    <Last name>GUINNESS</Last name>
14    <actorId>1</actorId>
15  </Actors>
16  <Actors>
17    <First name>CHRISTIAN</First na
18    <Last name>GABLE</Last name>
19    <actorId>10</actorId>
20  </Actors>
21  <Actors>
22    <First name>LUCILLE</First name
23    <Last name>TRACY</Last name>
24    <actorId>20</actorId>
25  </Actors>
26  <Actors>
27    <First name>SANDRA</First name>
28    <Last name>PECK</Last name>
```

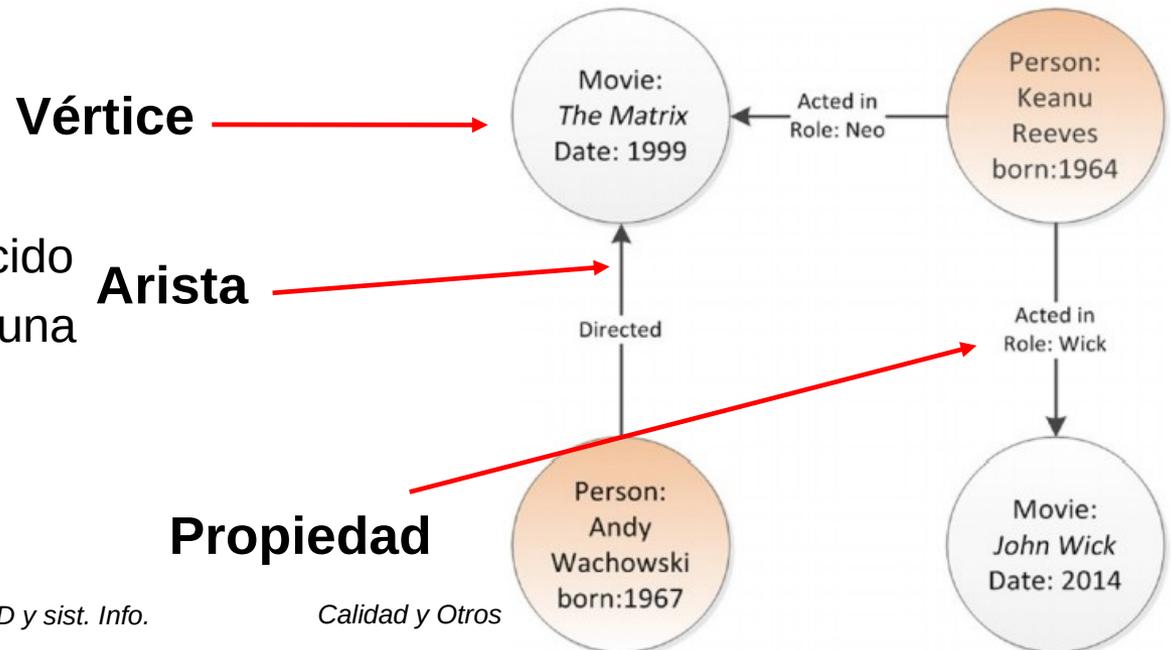
Bases de Datos de Documentos

- *MongoDB*, de humongous (gigantesco), es una BD orientada a documentos multi-plataforma. Evita la tradicional estructura de BDR basada en tablas utilizando documentos similares a JSON (llamada BSON), con esquemas dinámicos.

Bases de Datos de grafos

- Almacenan estructuras de datos que tienen topología de grafo.
- Se basan en la teoría de grafos, la cual define los principales componentes de un grafo:
 - Vértices o nodos, que representan distintos objetos
 - Aristas, relaciones o arcos que conectan los objetos
 - Vértices y aristas pueden tener propiedades

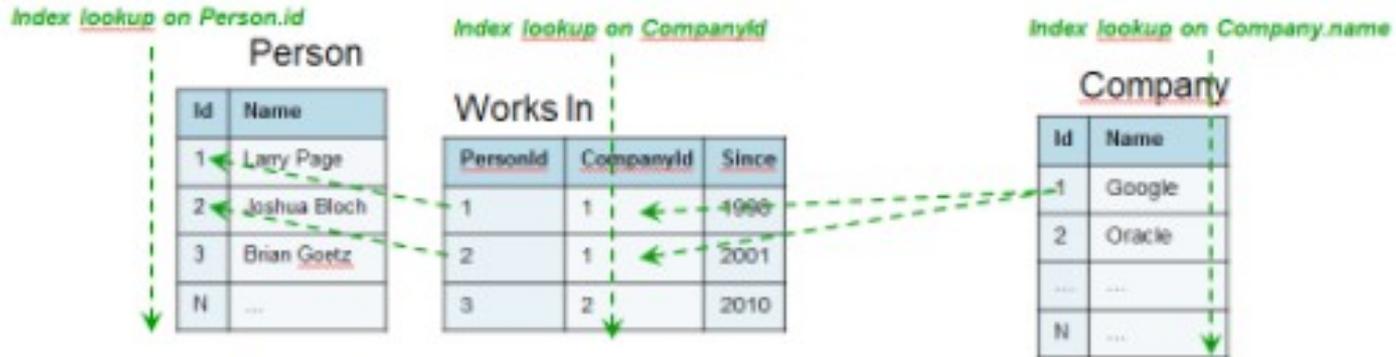
Podríamos realizar un recorrido en el grafo para encontrar todos los actores que han aparecido como co-protagonistas en una película protagonizada por Keanu Reeves.



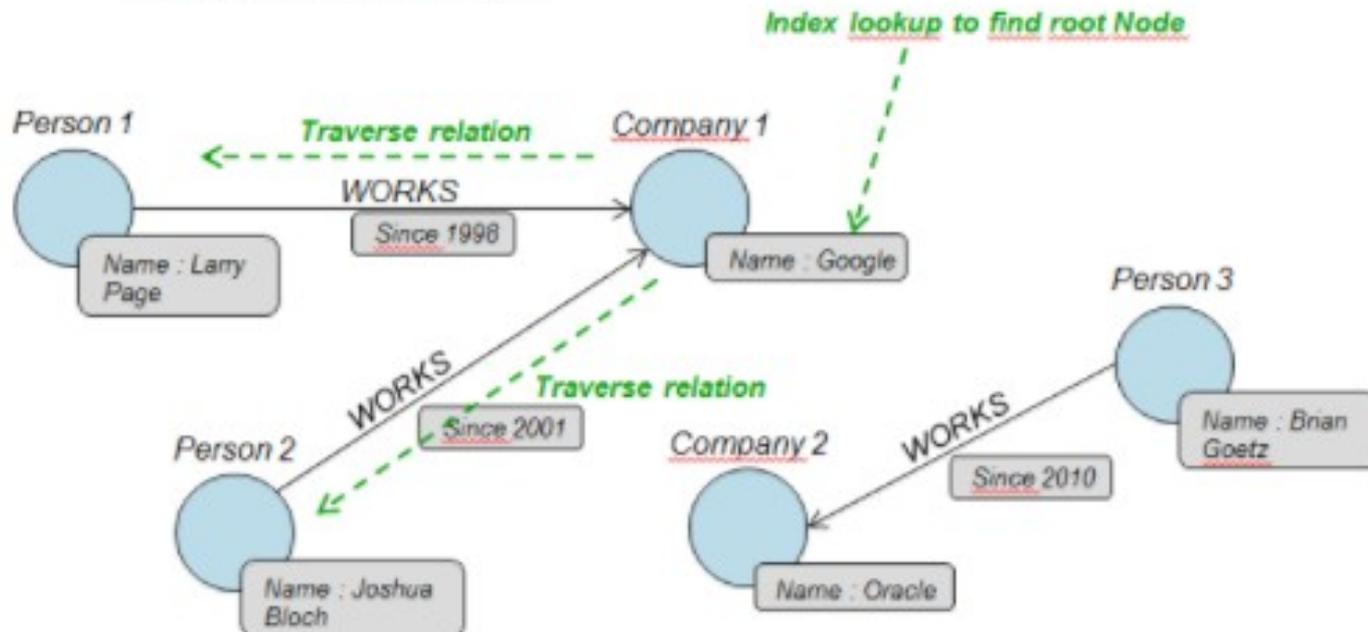
Bases de Datos de grafos

- Algunas bases de datos de grafos son internamente *key-value stores*, que agregan el concepto de las relaciones entre registros.
- El modelo de datos es flexible, lo que en algunos casos nos permite NO declarar tipos de vértices o de aristas.
 - El tipo de vértice o arista sería equivalente a las distintas tablas que se definen en un modelo relacional.
- Este tipo de BD facilita, en general, la exploración de datos, especialmente cuando las relaciones entre esos datos son tan significativas como los datos mismos.
 - Las BDR están diseñadas para ejecutar consultas globales, gracias a la estructura interna de tablas. Están poco preparadas para explorar relaciones de forma masiva, donde necesitan usar claves foráneas accediendo a tablas intermedias.

Comparando BDR y Grafos



Select Person Name
from Person, Company, WorksIn
where Company.name='Google'
and WorksIn.CompanyId = Company.Id
and WorksIn.PersonId = Person.Id



Bases de Datos de grafos: RDF

- RDF (*Resource Description Framework*) es una familia de especificaciones de la World Wide Web Consortium (W3C) originalmente diseñado como un modelo de datos para metadatos (datos de datos).
- RDF es un estándar web desarrollado a fines de los 90 para modelar recursos web y las relaciones entre ellos. Representa uno de los primeros estándares para representar y procesar los datos de grafos.

Bases de Datos de grafos: RDF

- La información en RDF se expresa en triplas, conceptualmente tiene la forma entidad, atributo y valor

- Por ejemplo:

TheMatrix: is :Movie

Kenau: is :Person

Kenau: starred in: TheMatrix

- Los grafos RDF se pueden almacenar en variedad de formatos:
 - Por ejemplo: XML, o incluso dentro de las tablas de una BDR.
- Sin embargo, se han implementado bases de datos RDF nativas, conocidas como *triplestores*.

Bases de Datos de grafos: RDF

- RDF soporta un lenguaje de consulta *SPARQL*, que es un lenguaje como SQL para consultar datos RDF.
- Ejemplo: consulta *SPARQL* sobre DBpedia (representación RDF de datos seleccionados de Wikipedia) para encontrar todos los objetos que tienen a Edgar Codd como un tema.

```
SELECT ?object
FROM <http://dbpedia.org>
WHERE { <http://dbpedia.org/resource/Edgar_F._Codd>
  <http://purl.org/dc/terms/subject> ?object }
```

| object |
|---|
| http://dbpedia.org/resource/Category:1923_births |
| http://dbpedia.org/resource/Category:2003_deaths |
| http://dbpedia.org/resource/Category:Alumni_of_Exeter_College_Oxford |
| http://dbpedia.org/resource/Category:British_computer_scientists |
| http://dbpedia.org/resource/Category:Cellular_automatists |
| http://dbpedia.org/resource/Category:Deaths_from_myocardial_infarction |
| http://dbpedia.org/resource/Category:Fellows_of_the_Association_for_Computing_Machinery |
| http://dbpedia.org/resource/Category:IBM_Fellows |
| http://dbpedia.org/resource/Category:IBM_employees |
| http://dbpedia.org/resource/Category:Turing_Award_laureates |
| http://dbpedia.org/resource/Category:University_of_Michigan_alumni |
| http://dbpedia.org/resource/Category:Database_researchers |
| http://dbpedia.org/resource/Category:People_educated_at_Poole_Grammar_School |
| http://dbpedia.org/resource/Category:People_from_San_Jose_California |

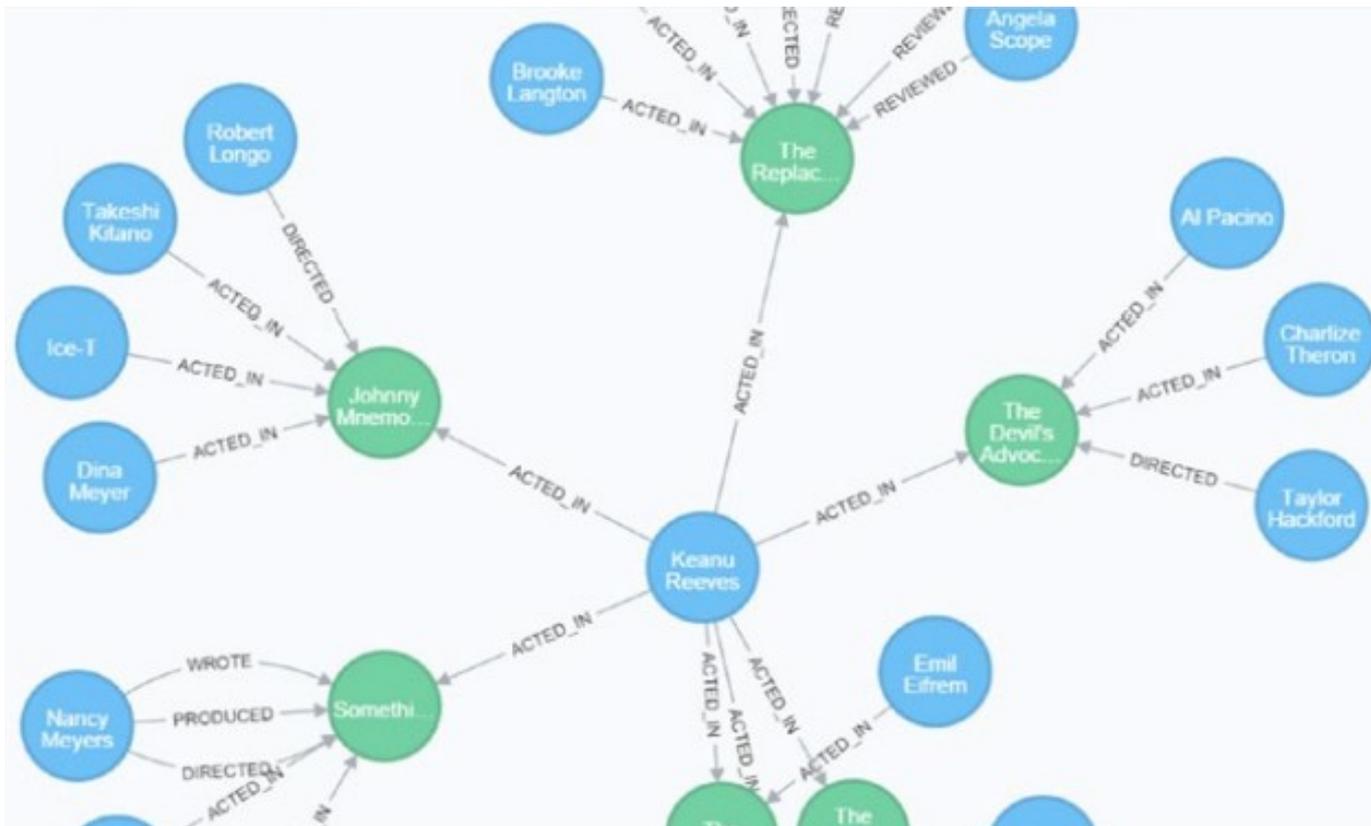
(Security restrictions of this...
Results Format:
Execution timeout:
Options:
(The result can only be sent...
Run Query Reset

Bases de Datos de grafos: Neo4j

- Neo4j es la base de datos de grafos más adoptada.
- Neo4j implementa un lenguaje de consulta de grafos declarativo: *Cypher*.
- Cypher permite que los grafos sean consultados utilizando una sintaxis simple comparable con SQL o SPARQL, pero particularmente optimiza los recorridos de los grafos.

Bases de Datos de grafos: Neo4j

- Cypher devuelve resultados que pueden ser grafos. Por ejemplo, la imagen es un grafo que muestra todos los vértices relacionados con "Keanu Reeves" a dos niveles.



Resumen

- Los sistemas de BD NoSQL crecieron con las principales redes sociales como Google, Amazon, Twitter y Facebook.
- Las BD NoSQL son una clase de DBMS que difieren del modelo clásico de RDBMS en aspectos importantes:
 - no usan SQL como lenguaje principal de consultas
 - los datos almacenados no requieren estructuras fijas como tablas
 - normalmente no soportan operaciones JOIN
 - no garantizan completamente ACID (atomicidad, consistencia, aislamiento y durabilidad)

Bibliografía

- Libro: Next Generation Databases:
https://link-springer-com.proxy.timbo.org.uy:88/chapter/10.1007/978-1-4842-1329-2_3
- MongoDB: <https://www.mongodb.com/>
- SPARQL: <http://www.sparql.org/>
- Cypher: <https://neo4j.com/developer/cypher-query-language/>
- W3C: <https://www.w3.org/>
- Tutorial: XML Databases for Bioinformatics:
https://psb.stanford.edu/previous/psb02/tut_xml.html
- Are graph databases ready for bioinformatics?:
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3842757/>
- Recon2Neo4j: <https://academic.oup.com/bioinformatics/article/33/7/1096/2557691/Recon2Neo4j-applying-graph-database-technologies>
- Bio4j: <http://bio4j.com/>