

Fig. 12.11 Ejemplo de tablas web/tablas de fusión

Tablas web. La infraestructura de la tabla de fusión puede detectar automáticamente el atributo de unión entre tablas y generar oportunidades de integración. La figura 12.11 muestra un ejemplo que muestra dos conjuntos de datos aportados por dos propietarios diferentes: uno sobre restaurantes y el otro sobre las puntuaciones y calificaciones otorgadas a estos restaurantes como resultado de una inspección. El sistema determinaría que ambos conjuntos de datos pueden unirse mediante un atributo común y proporcionar acceso integrado. Si bien en este caso ambas tablas fueron aportadas por los usuarios, en otros casos, una o ambas tablas pueden detectarse desde la web utilizando las técnicas desarrolladas por el proyecto de tablas web.

12.6.2 Web semántica y datos abiertos enlazados

Una contribución fundamental de la web es la creación de un repositorio de datos procesables por máquinas . Semanticweb busca convertir estos datos en un formato comprensible para las máquinas mediante la integración de datos estructurados y no estructurados en la web y su marcado semántico. La visión original de la web semántica incluye tres componentes:

- Marcar datos web para que los metadatos se capturen como anotaciones; •
- Utilizar ontologías para hacer que las diferentes colecciones de datos sean comprensibles; y
- Utilizar tecnologías basadas en lógica para acceder tanto a los metadatos como a las ontologías.

Los Datos Abiertos Vinculados (LOD) se introdujeron en 2006 como una aclaración de esta visión, enfatizando los vínculos entre los datos que forman parte de la web semántica. Establecieron directrices sobre cómo publicar los datos en la web para lograr la visión de la web semántica. Por lo tanto, la web semántica es una visión de integración de datos web realizada mediante LOD. Los requisitos de LOD para publicar (y, por consiguiente, integrar) datos en la web se basan en cuatro principios:

- Todos los recursos web (datos) se identifican localmente mediante sus URI, que sirven como nombres;
- Estos nombres son accesibles mediante HTTP;
- La información sobre los recursos/entidades web se codifica como tripletes RDF (Resource Description Framework). En otras palabras, RDF es el modelo de datos de la web semántica (y lo explicamos más adelante).
- Las conexiones entre conjuntos de datos se establecen mediante enlaces de datos y los editores de conjuntos de datos deben establecer estos enlaces para que se puedan descubrir más datos.

Por lo tanto, el LOD genera un grafo donde los vértices son recursos web y las aristas, las relaciones. En la Fig. 12.12 se muestra una forma simplificada del "grafo LOD" de 2018, donde cada vértice representa un conjunto de datos (no un recurso web) categorizado por color (p. ej., publicaciones, ciencias de la vida, redes sociales) y el tamaño de cada vértice representa su grado de entrada. En ese momento, el LOD constaba de 1234 conjuntos de datos con 16 136 enlaces. Volveremos al tema del LOD y del grafo LOD en breve.

La web semántica consta de varias tecnologías que se complementan entre sí (Fig. 12.13). En la capa inferior, XML proporciona el lenguaje para escribir documentos web estructurados e intercambiarlos fácilmente. Por encima de esto se encuentra el RDF que, como se mencionó anteriormente, establece el modelo de datos. Aunque no es necesario, si se especifica un esquema sobre estos datos, el esquema RDF proporciona las primitivas necesarias. Las ontologías amplían el esquema RDF con construcciones más potentes para especificar las relaciones entre los datos web. Finalmente, los lenguajes de reglas declarativas basados en la lógica permiten que las aplicaciones definan sus propias reglas.

En el resto discutimos las tecnologías en las capas inferiores, ya que son los requisitos mínimos.

### 12.6.2.1 XML

La codificación predominante para documentos web ha sido HTML (lenguaje de marcado de hipertexto). Un documento web codificado en HTML consta de elementos HTML encapsulados por etiquetas, como se explicó en la sección 12.6.1, donde también presentamos enfoques para descubrir datos estructurados en documentos web codificados en HTML e integrarlos. Como se mencionó anteriormente, en el contexto de la web semántica, la representación preferida para codificar e intercambiar documentos web es XML.

---

<sup>5</sup>Estadísticas obtenidas de <https://lod-cloud.net>, que debe consultarse para obtener estadísticas actualizadas.

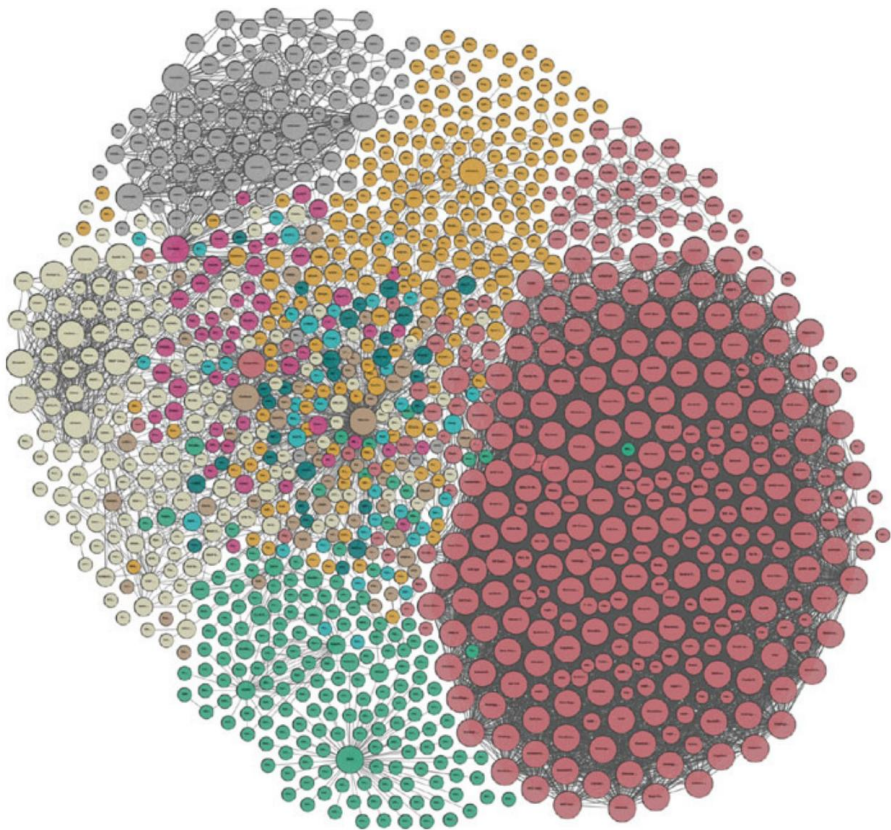


Fig. 12.12 Gráfico LOD a partir de 2018

Declarativo
Lenguajes de reglas
Lenguajes de ontología
Esquema RDF
RDF
XML

Fig. 12.13 Tecnologías de la web semántica. Simplificado de [Antoniou y Plexousakis, 2018]

(que significa Lenguaje de Marcado Extenso) propuesto por el Consorcio World Wide Web (W3C).

Las etiquetas XML (también llamadas marcados) dividen los datos en fragmentos llamados elementos, con el objetivo de proporcionar mayor semántica a los datos. Los elementos pueden anidarse, pero no superponerse. La anidación de elementos representa relaciones jerárquicas entre ellos. A modo de ejemplo, la figura 12.14 muestra la representación XML, con ligeras modificaciones, de los datos bibliográficos que presentamos anteriormente.

Un documento XML se puede representar como un trie que contiene un elemento raíz, el cual tiene cero o más subelementos anidados (o elementos secundarios), que pueden contener subelementos recursivamente. Para cada elemento, hay cero o más atributos con valores atómicos asignados. Un elemento también contiene un valor opcional. Debido a la representación textual del trie, se define un orden total, denominado orden del documento, para todos los elementos, que corresponde al orden en que aparece el primer carácter de los elementos en el documento.

Por ejemplo, el elemento raíz de la Fig. 12.4 es bib, que tiene tres elementos secundarios: dos libros y un artículo. El primer elemento, "book", tiene un atributo "year" con valor atómico "1999" y también contiene subelementos (p. ej., el elemento "title"). Un elemento puede contener un valor (p. ej., "Principles of Distributed Database Systems" para el elemento "title").

La definición de un documento XML estándar es un poco más compleja: puede contener ID-IRDREFs, que definen referencias entre elementos del mismo documento o de otro. En ese caso, la representación del documento se convierte en un grafo.

Sin embargo, es bastante común utilizar la representación trie más simple, y asumiremos lo mismo en esta sección y la definiremos con más precisión a continuación.<sup>6</sup> Un

documento XML se modela como un trie ordenado y etiquetado por nodos  $T = (V, E)$ , donde cada nodo  $v \in V$  corresponde a un elemento o atributo y se caracteriza por:

- un identificador único denotado por  $ID(v)$ ; • una propiedad de tipo única, denotada como  $kind(v)$ , asignada desde el conjunto {elemento, atributo, texto}; • una etiqueta, denotada por  $label(v)$ , asignada desde algún alfabeto; • un contenido, denotado por  $content(v)$ , que está vacío para nodos que no son hojas y es un

Fuente para los nodos de las hojas.

Una arista dirigida  $e = (u, v)$  está incluida en  $E$  si y solo si:

- $kind(u) = kind(v) = \text{elemento}$ , y  $v$  es un subelemento de  $u$ ; o •  $kind(u) = \text{elemento}$   $kind(v) = \text{atributo}$ , y  $v$  es un atributo de  $u$ .

Ahora que un trie de documento XML está correctamente definido, podemos definir una instancia de modelo de datos XML como una colección ordenada (secuencia) de nodos o valores atómicos de un trie de documento XML. Un esquema puede o no estar definido para un documento XML, ya que es un formato autodescriptivo. Si se define un esquema para una colección de

<sup>6</sup>Además, omitimos los nodos de comentarios, los nodos de espacio de nombres y los nodos PI del modelo.

```

<bib>
<book year = "1999">
<author> M. Tamer Ozsu </author>
<author> Patrick Valduriez </author>
<title> Principles of Distributed ... </title>
<chapters>
<chapter>
<heading> ... </heading>
<body> ... </body>
</chapter>
...
<chapter>
<heading> ... </heading>
<body> ... </body>
</chapter>
</chapters>
<price currency= "USD"> 98.50 </price>
</book>
<article year = "2009">
<author> M. Tamer Ozsu </author>
<author> Yingying Tao </author>
<title> Mining data streams ... </title>
<venue> "CIKM" </venue>
<sections>
<section> ... </section>
...
<section> ... </section>
</sections>
</article>
<book>
<author> Anthony Bonato </author>
<title> A Course on the Web Graph </title>
<ISBN> TK5105.888.B667 </ISBN>
<chapters>
<chapter>
<heading> ... </heading>
<body> ... </body>
</chapter>
<chapter>
<heading> ... </heading>
<body> ... </body>
</chapter>
<chapter>
<heading> ... </heading>
<body> ... </body>
</chapter>
</chapters>
<publisher> AMS </publisher>
</book>
</bib>

```

Fig. 12.14 Un ejemplo de documento XML

Documentos XML: cada documento de esta colección se ajusta a dicho esquema; sin embargo, este permite variaciones en cada documento, ya que no todos los elementos o atributos pueden existir en cada uno. Los esquemas XML se pueden definir mediante la Definición de Tipo de Documento (DTD) o XMLSchema. En esta sección, utilizaremos una definición de esquema más sencilla que aprovecha la estructura gráfica de los documentos XML, como se definió anteriormente.

Un gráfico de esquema XML se define como una tupla de 5  $(\Sigma, \mathcal{N}, \mathcal{E}, s, m, p)$  donde  $\Sigma$  es un alfabeto de tipos de nodos de documentos XML,  $p$  es el tipo de nodo raíz,  $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$  es un conjunto de aristas entre tipos de nodos,  $s : \mathcal{E} \rightarrow \{\text{ONCE}, \text{OPT}, \text{MULT}\}$  y  $m : \mathcal{N} \rightarrow \{\text{string}\}$ . La semántica de esta definición es la siguiente: Una arista  $\psi = (\sigma_1, \sigma_2) \in \mathcal{E}$  denota que un elemento de tipo  $\sigma_1$  puede contener un elemento de tipo  $\sigma_2$ .  $s(\psi)$  denota la cardinalidad de la contención representada por esta arista: Si  $s(\psi) = \text{ONCE}$ , entonces un elemento de tipo  $\sigma_1$  debe contener exactamente un elemento de tipo  $\sigma_2$ . Si  $s(\psi) = \text{OPT}$ , entonces un elemento de tipo  $\sigma_1$  puede o no contener un elemento de tipo  $\sigma_2$ . Si  $s(\psi) = \text{MULT}$ , entonces un elemento de tipo  $\sigma_1$  puede contener múltiples elementos de tipo  $\sigma_2$ .  $m(\sigma)$  denota el dominio del contenido de texto de un elemento de tipo  $\sigma$ , representado como el conjunto de todas las cadenas que pueden aparecer dentro de dicho elemento.

Utilizando la definición del modelo de datos XML y las instancias de este modelo, ahora es posible definir los lenguajes de consulta. Las expresiones en lenguajes de consulta XML toman una instancia de datos XML como entrada y producen otra instancia de datos XML como salida. XPath y XQuery son dos lenguajes de consulta propuestos por el W3C. Las expresiones de ruta, que presentamos anteriormente, están presentes en ambos lenguajes de consulta y son posiblemente la forma más natural de consultar datos XML jerárquicos. XQuery define construcciones más potentes. Aunque XQuery fue objeto de una intensa investigación y desarrollo en la década del 2000, su uso ya no está muy extendido.

Es complicado, difícil de formular por los usuarios y difícil de optimizar por los sistemas.

JSON ha reemplazado tanto a XML como a XQuery para muchas aplicaciones, como analizamos en el Capítulo 11, aunque la representación XML sigue siendo importante para la web semántica (pero no para XQuery).

### 12.6.2.2 RDF

RDF es el modelo de datos sobre XML y constituye un componente fundamental de la web semántica (Fig. 12.13). Aunque fue propuesto originalmente por el W3C como un componente de la web semántica, su uso ahora es más amplio. Por ejemplo, Yago y DBPedia extraen datos de Wikipedia automáticamente y los almacenan en formato RDF para admitir consultas estructurales en Wikipedia; los biólogos codifican sus experimentos y resultados utilizando RDF para comunicarse entre sí, lo que da lugar a colecciones de datos RDF, como Bio2RDF ([bio2rdf.org](http://bio2rdf.org)) y Uniprot RDF ([dev.isb-sib.ch/projects/uniprot-rdf](http://dev.isb-sib.ch/projects/uniprot-rdf)). Relacionado con la web semántica, el proyecto LOD construye una nube de datos RDF vinculando un gran número de conjuntos de datos, como se mencionó anteriormente.

RDF modela cada "hecho" como un conjunto de triples (sujeto, propiedad (o predicado), objeto), denotados como  $s, p, o$ , donde sujeto es una entidad, clase o nodo en blanco, un

La propiedad<sup>7</sup> denota un atributo asociado a una entidad, y el objeto es una entidad, una clase, un nodo vacío o un valor literal. Según el estándar RDF, una entidad se denota mediante un URI (Identificador Uniforme de Recursos) que hace referencia a un recurso con nombre en el entorno que se modela. Los nodos vacíos, en cambio, hacen referencia a recursos anónimos sin nombre.<sup>8</sup> Por lo tanto, cada tripleta representa una relación con nombre; las que incluyen nodos vacíos simplemente indican que existe algo con la relación dada, sin nombrarlo.

En este punto, conviene hablar brevemente sobre la siguiente capa de la pila tecnológica de la web semántica (Fig. 12.13), el esquema RDF (RDFS). Es posible anotar datos RDF con metadatos semánticos mediante RDFS, que también es un estándar del W3C.<sup>9</sup> Esta anotación permite principalmente el razonamiento sobre los datos RDF (denominado implicación) y, en algunos casos, también influye en la organización de los datos. Los metadatos pueden utilizarse para la optimización de consultas semánticas. Ilustramos los conceptos fundamentales con ejemplos sencillos utilizando RDFS, que permite la definición de clases y jerarquías de clases. RDFS cuenta con definiciones de clase integradas; las más importantes son `rdfs:Class` y `rdfs:subClassOf`, que se utilizan para definir una clase y una subclase, respectivamente ( en los ejemplos de consulta a continuación se utiliza `rdfs:label` ). Para especificar que un recurso individual es un elemento de la clase, se utiliza una propiedad especial, `rdf:type`.

Ejemplo 12.13 Por ejemplo, si quisiéramos definir una clase llamada Películas y dos subclases Películas de Acción y Dramas, esto se lograría de la siguiente manera:

```
Películas rdf:type rdfs:Class .
Películas de acción rdfs:subClassOf Películas .
Dramas rdfs:subClassOf Películas .
```

Formalmente, un conjunto de datos RDF se puede definir de la siguiente manera. Sean  $U$ ,  $B$ ,  $L$  y  $V$  los conjuntos de todas las URI, nodos vacíos, literales y variables, respectivamente. Una tupla  $(s, p, o)$   $(U \cup B) \times U \times (U \cup B \cup L)$  es una terna RDF. Un conjunto de ternas RDF forma un conjunto de datos RDF.

Ejemplo 12.14 En la figura 12.15 se muestra un ejemplo de conjunto de datos RDF, donde los datos provienen de varias fuentes según lo definido por los prefijos URI.

<sup>7</sup>En la literatura, los términos "propiedad" y "predicado" se utilizan indistintamente; en este artículo, utilizaremos "propiedad" de manera consistente.

En gran parte de la investigación, se ignoran los nodos vacíos. Salvo que se indique explícitamente lo contrario, también los ignoraremos en este artículo.

La misma anotación también se puede realizar utilizando lenguajes de ontología como OWL (también un estándar del W3C), pero no discutiremos ese tema más a fondo.

Prefijos:

```
mdb=http://data.linkedmdb.org/resource/geo=http://sws.geonames.org/ bm=http://
wifo5-03.informatik.uni-mannheim.de/bookmashup/ exvo=http://lexvo.org/id/
wp=http://en.wikipedia.org/wiki/
```

Subject	Property	Object
mdb: film/2014	rdfs:label	"The Shining"
mdb:film/2014	movie:initial_release_date	"1980-05-23"
mdb:film/2014	movie:director	mdb:director/8476
mdb:film/2014	movie:actor	mdb:actor/29704
mdb:film/2014	movie:actor	mdb: actor/30013
mdb:film/2014	movie:music_contributor	mdb: music_contributor/4110
mdb:film/2014	foaf:based_near	geo:2635167
mdb:film/2014	movie:relatedBook	bm:0743424425
mdb:film/2014	movie:language	lexvo:iso639-3/eng
mdb:director/8476	movie:director_name	"Stanley Kubrick"
mdb:film/2685	movie:director	mdb:director/8476
mdb:film/2685	rdfs:label	"A Clockwork Orange"
mdb:film/424	movie:director	mdb:director/8476
mdb:film/424	rdfs:label	"Spartacus"
mdb:actor/29704	movie:actor_name	"Jack Nicholson"
mdb:film/1267	movie:actor	mdb:actor/29704
mdb:film/1267	rdfs:label	"The Last Tycoon"
mdb:film/3418	movie:actor	mdb:actor/29704
mdb:film/3418	rdfs:label	"The Passenger"
geo:2635167	gn:name	"United Kingdom"
geo:2635167	gn:population	62348447
geo:2635167	gn:wikipediaArticle	wp:United_Kingdom
bm:books/0743424425	dc:creator	bm:persons/Stephen+King
bm:books/0743424425	rev:rating	4.7
bm:books/0743424425	scom:hasOffer	bm:offers/0743424425amazonOffer
lexvo:iso639-3/eng	rdfs:label	"English"
lexvo:iso639-3/eng	lvont:usedIn	lexvo:iso3166/CA
lexvo:iso639-3/eng	lvont:usesScript	lexvo:script/Latn

Fig. 12.15 Ejemplo de conjunto de datos RDF. Se utilizan prefijos para identificar las fuentes de datos.

Los datos RDF se pueden modelar como un grafo RDF de la siguiente manera. Un grafo RDF es una sextupla  $G = V, LV, fV, E, LE, fE$ , donde

- 1.  $V = V_c \cup V_e \cup V_l$  es una colección de vértices que corresponden a todos los sujetos y objetos en datos RDF, donde  $V_c$ ,  $V_e$  y  $V_l$  son colecciones de vértices de clase, vértices de entidad y vértices literales, respectivamente.
- 2.  $LV$  es una colección de etiquetas de vértices.
- 3. Una función de etiquetado de vértices  $fV : V \rightarrow LV$  es una función biyectiva que asigna una etiqueta a cada vértice. La etiqueta de un vértice  $u \in V_l$  es su valor literal, y la etiqueta de un vértice  $u \in V_c \cup V_e$  es su URI correspondiente.
- 4.  $E = \{-u \rightarrow v, u \rightarrow v\}$  es una colección de aristas dirigidas que conectan los correspondientes sujetos y objetos.
- 5.  $LE$  es una colección de etiquetas de borde.



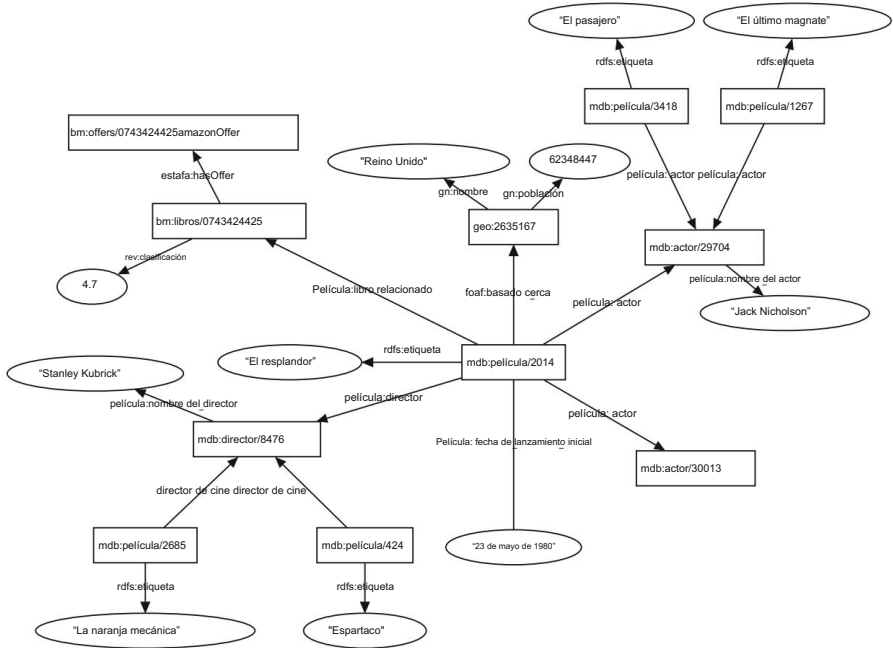


Fig. 12.16 Gráfico RDF correspondiente al conjunto de datos de la Fig. 12.15

6. Una función de etiquetado de aristas  $f_E : E \rightarrow LE$  es una función biyectiva que asigna una etiqueta a cada arista. La etiqueta de una arista  $e \in E$  es su propiedad correspondiente.

Una arista  $-u \xrightarrow{p} v$  es una arista de propiedad de atributo si  $v$  es un literal; de lo contrario, es una arista de enlace.

Tenga en cuenta que la estructura del gráfico RDF es diferente a la de los gráficos de propiedades que analizamos en el capítulo 10. Como recordará, los gráficos de propiedades tienen atributos asociados a vértices y aristas que permiten especificar predicados sofisticados basados en valores en las consultas.

En los gráficos RDF, el único atributo de un vértice o una arista es la etiqueta del vértice/arista.

Lo que serían atributos de vértice en un grafo de propiedades se convierten en aristas cuyas etiquetas son los nombres de los atributos. Por lo tanto, los grafos RDF son más simples y regulares, pero generalmente más grandes en cuanto a número de vértices y aristas.

La Figura 12.16 muestra un ejemplo de un grafo RDF. Los vértices que se indican en los cuadros por caja son vértices de entidad o clase, y los demás son vértices literales.

El lenguaje estándar del W3C para RDF es SPARQL, que se define de la siguiente manera [Hartig 2012]. Sean  $U, B, L$  y  $V$  los conjuntos de todas las URI, nodos en blanco, literales y variables, respectivamente. Una expresión SPARQL se expresa recursivamente.

1. Un patrón triple  $(U \ B \ V) \times (U \ V) \times (U \ B \ L \ V)$  es una expresión SPARQL. 2. (opcionalmente) Si  $P$  es una expresión SPARQL, entonces  $P \text{ FILTER } R$  también es una expresión SPARQL donde  $R$  es una condición de filtro SPARQL incorporada.

3. (opcionalmente) Si P1 y P2 son expresiones SPARQL, entonces P1 AND|OPT |OR P2 También son expresiones SPARQL.

Un conjunto de patrones triples se denomina patrón de gráfico básico (BGP) y SPARQL. Las expresiones que solo contienen estos se denominan consultas BGP. Estas son el tema de la mayor parte de la investigación en evaluación de consultas SPARQL.

Ejemplo 12.15 Un ejemplo de consulta SPARQL que encuentra los nombres de las películas dirigida por "Stanley Kubrick" y tiene un libro relacionado que tiene una calificación mayor a 4.0 se especifica de la siguiente manera:

```
SELECCIONAR ?nombre
DÓNDE {
  ?m rdfs:etiqueta ?nombre. ?m película:director ?d.
  ?d película:nombre_director "Stanley Kubrick".
  ?m película:libro relacionado ?b. ?b rev:clasificación ?r.
  FILTER(?r > 4.0)
}
```

En esta consulta, las primeras tres líneas de la cláusula WHERE forman un BGP que consta de Cinco patrones triples. Todos los patrones triples de este ejemplo tienen variables, como "?m", "?name" y "?r", y "?r" tiene un filtro: FILTER(?r > 4.0).

Una consulta SPARQL también se puede representar como un gráfico de consulta. Un gráfico de consulta es un séptuple  $Q = V, EQ, LQ, f, f_{mi}, FL$ , donde

- $V$  es una colección de vértices que corresponden a todos sujetos y objetos en una consulta SPARQL, donde  $V$  es una colección de variables vértices (correspondientes a variables en la expresión de consulta) y  $V$  y  $V$  son colecciones de vértices de clase, vértices de entidad y vértices literales en el gráfico de consulta  $Q$ , respectivamente.
- $EQ$  es una colección de aristas que corresponden a propiedades en una consulta SPARQL.
- $LQ$  es una colección de etiquetas de vértices en  $Q$  y  $LQ$  son las etiquetas de borde en  $EQ$ .
- $f$  es una función de etiquetado de vértices biyectiva que asigna a cada vértice en  $Q$  una etiqueta de  $LQ$ . La etiqueta de un vértice  $v$  es la variable; que de un vértice  $v$  es su valor literal; y el de un vértice  $v$  es su correspondiente.
- $f_{mi}$  es una función de etiquetado de vértices biyectiva que asigna a cada borde en  $Q$  una etiqueta de  $LQ$ . Una etiqueta de borde puede ser una propiedad o una variable de borde.
- $FL$  son filtros de restricción.

El gráfico de consulta para  $Q_1$  se muestra en la figura 12.17.

Por lo tanto, la semántica de la evaluación de consultas SPARQL se puede definir como Coincidencia de subgrafos mediante homomorfismo de grafos, por el cual todos los subgrafos de un RDF Se encuentran gráficos  $G$  que son homomórficos al gráfico de consulta SPARQL  $Q$ .

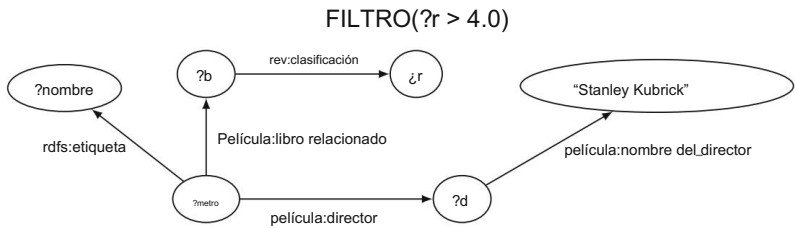


Fig. 12.17 Gráfico de consulta SPARQL correspondiente a la consulta Q1

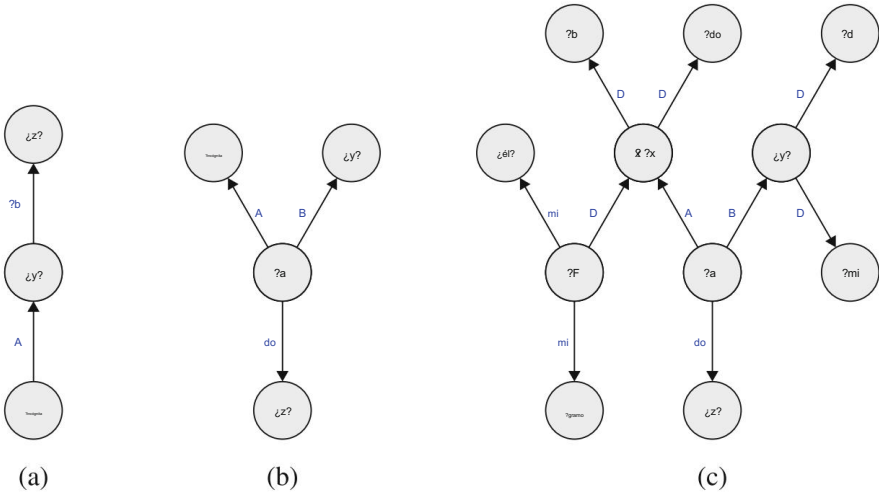


Fig. 12.18 Formas de consulta SPARQL de muestra. (a) QL. (b) QS . (c) QK

Es habitual hablar de tipos de consulta SPARQL según la forma del grafo de consulta (nos referiremos a estos tipos en la siguiente discusión). Típicamente, se observan tres tipos de consulta: (i) lineal (Fig. 12.18a), donde la variable en el campo de objeto de un patrón triple aparece en el sujeto de otro patrón triple (p. ej., ?y en QL); (ii) en forma de estrella (Fig. 12.18b), donde la variable en el campo de objeto de un patrón triple aparece en el sujeto de varios otros patrones triples (p. ej., ?a en QS); y (iii) en forma de copo de nieve (Fig. 12.18c), que es una combinación de múltiples consultas en estrella.

Se han desarrollado diversos sistemas de gestión de datos RDF. Estos se pueden clasificar, a grandes rasgos, en cinco grupos: aquellos que asignan los datos RDF directamente a un sistema relacional, aquellos que utilizan un esquema relacional con indexación extensa (y un sistema de almacenamiento nativo), aquellos que desnormalizan la tabla de triples en propiedades agrupadas, aquellos que utilizan la organización de almacenamiento en columnas y aquellos que aprovechan la semántica nativa de coincidencia de patrones de grafos de SPARQL.

### Mapeo relacional directo

Los sistemas de mapeo relacional directo aprovechan la estructura tabular natural de las tripletas RDF. Por lo tanto, crean una única tabla con tres columnas (Asunto, Propiedad, Objeto) que contiene las tripletas (normalmente existen tablas auxiliares adicionales, pero las ignoramos aquí). La consulta SPARQL puede entonces traducirse a SQL y ejecutarse en esta tabla. Se ha demostrado que SPARQL 1.0 puede traducirse completamente a SQL; aún queda por determinar si esto mismo ocurre con SPARQL 1.1 y sus características adicionales. Este enfoque busca aprovechar las técnicas bien desarrolladas de almacenamiento relacional, procesamiento de consultas y optimización para la ejecución de consultas SPARQL. Sistemas como Sesame SQL92SAIL<sup>10</sup> y Oracle siguen este enfoque.

Ejemplo 12.16 Suponiendo que la tabla dada en la Fig. 12.15 es una tabla relacional, la consulta SPARQL de ejemplo en el Ejemplo 12.15 se puede traducir a la siguiente consulta SQL (donde s, p, o corresponden a los nombres de columna: Sujeto, Propiedad, Objeto):

```
SELECCIONAR T1.object
DE T COMO T1, T COMO T2, T COMO T3, T COMO
T4, T COMO T5 DONDE
T1.p="rdfs:label"
Y T2.p="película:librorelacionado"
Y T3.p="película:director"
Y T4.p="rev:clasificación"
Y T5.p="película:nombre_del_director"
Y T1.s=T2.s
Y T1.s=T3.s
Y T2.o=T4.s
Y T3.o=T5.s
Y T4.o > 4.0
Y T5.o="Stanley Kubrick"
```

Como se puede observar en este ejemplo, este enfoque genera un gran número de autouniones que no son fáciles de optimizar. Además, en conjuntos de datos grandes, esta tabla de triples individuales se vuelve muy extensa, lo que complica aún más el procesamiento de consultas.

---

<sup>10</sup>Sesame está diseñado para interactuar con cualquier sistema de almacenamiento, ya que implementa una Capa de Almacenamiento e Inferencia (SAIL) para interactuar con el sistema de almacenamiento específico en el que se encuentra. SQL92SAIL es la instancia específica para trabajar en sistemas relacionales.

### Indexación extensa de una sola tabla

Una alternativa a los problemas que genera el mapeo relacional directo es desarrollar sistemas de almacenamiento nativos que permitan la indexación exhaustiva de la tabla triple. Hexastore y RDF-3X son ejemplos de este enfoque. La tabla única se mantiene, pero se indexa exhaustivamente. Por ejemplo, RDF-3X crea índices para las seis permutaciones posibles del sujeto, la propiedad y el objeto: (spo, sop, ops, opo, pos, oos). Cada uno de estos índices se ordena lexicográficamente por la primera columna, seguida de la segunda y finalmente de la tercera. Estos se almacenan en las páginas hoja de un árbol B+ agrupado.

La ventaja de este tipo de organización es que las consultas SPARQL se pueden procesar eficientemente independientemente de dónde se encuentren las variables (sujeto, propiedad, objeto), ya que uno de los índices será aplicable. Además, permite el procesamiento de consultas basado en índices, eliminando algunas autouniones, que se convierten en consultas de rango sobre el índice específico. Incluso cuando se requieren uniones, se puede utilizar una rápida combinación de datos, ya que cada índice se ordena por la primera columna. Las desventajas obvias son, por supuesto, el uso de espacio y la sobrecarga de actualizar los múltiples índices si los datos son dinámicos.

### Tablas de propiedades

El enfoque de tablas de propiedades aprovecha la regularidad de los conjuntos de datos RDF, donde se repiten patrones de sentencias. Por consiguiente, almacena propiedades relacionadas en la misma tabla. El primer sistema que propuso este enfoque fue Jena; DB2RDF de IBM también sigue la misma estrategia. En ambos casos, las tablas resultantes se asignan a un sistema relacional y las consultas se convierten a SQL para su ejecución.

Jena define dos tipos de tablas de propiedades. El primer tipo, denominado tabla de propiedades agrupadas, agrupa las propiedades que suelen aparecer en los mismos sujetos (o similares). Define diferentes estructuras de tabla para propiedades de un solo valor y propiedades de varios valores. Para las propiedades de un solo valor, la tabla contiene la columna de sujeto y varias columnas de propiedad (Fig. 12.19a). El valor de una propiedad dada puede ser nulo si no existe una tripleta RDF que utilice el sujeto y dicha propiedad. Cada fila de la tabla representa varias tripletes RDF (la misma cantidad que los valores de propiedad no nulos). Para estas tablas, el sujeto es la clave principal. Para las propiedades de varios valores, la estructura de la tabla incluye el sujeto y la propiedad de varios valores (Fig. 12.19b). Cada fila de esta tabla representa una única tripleta RDF; la clave de la tabla es la clave compuesta (sujeto, propiedad). La asignación de la tabla de tripleta única a las tablas de propiedades es un problema de diseño de bases de datos que realiza el administrador de bases de datos.

Jena también define una tabla de clases de propiedades que agrupa a los sujetos con el mismo tipo de propiedad en una sola tabla de propiedades (Fig. 12.19c). En este caso, todos los miembros de una clase (recuerde nuestra discusión sobre la estructura de clases en el contexto de RDFS) se agrupan.

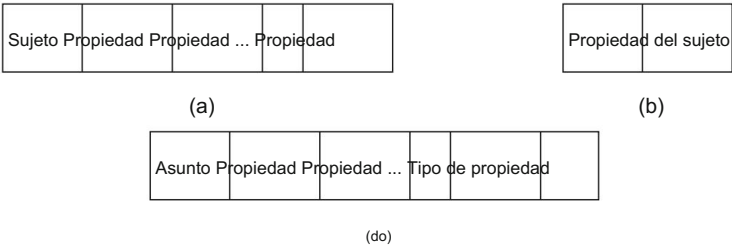


Fig. 12.19 Diseño de tabla de propiedades agrupadas

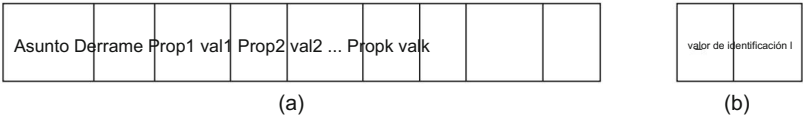


Figura 12.20 Diseño de tabla DB2RDF. (a) DPH. (b) DS

En una tabla. La columna "Tipo" contiene el valor de rdf:type para cada propiedad. fila.

Ejemplo 12.17 El conjunto de datos de ejemplo del Ejemplo 12.14 se puede organizar para crear una tabla que incluya las propiedades de los temas que son películas, una tabla para las propiedades de los directores, una tabla para las propiedades de los actores, una tabla para las propiedades de los libros, etcétera.

IBM DB2RDF sigue la misma estrategia, pero con una organización de tablas más dinámica (Fig. 12.20). La tabla, denominada hash primario directo (DPH), se organiza por cada objeto. Sin embargo, en lugar de identificar manualmente propiedades similares, alberga k columnas de propiedades, cada una de las cuales puede tener asignada una propiedad diferente en filas distintas. Cada columna de propiedad consta de dos columnas: una que contiene la etiqueta de la propiedad y la otra que contiene el valor. Si el número de propiedades de un objeto determinado es mayor que k, las propiedades adicionales se distribuyen en una segunda fila, que se marca en la columna de distribución. Para las propiedades multivalor, se mantiene una tabla hash secundaria directa (DSH): el valor original de la propiedad almacena un identificador único (I\_id), que aparece en la tabla DS junto con los valores.

La ventaja del enfoque de tabla de propiedades es que las uniones en las consultas en estrella (es decir, uniones sujeto-sujeto) se convierten en escaneos de tabla únicos. Por lo tanto, la consulta traducida tiene menos uniones. Las desventajas son que, en cualquiera de las dos formas mencionadas, podría haber un número significativo de valores nulos en las tablas (véase el número de valores nulos en la Fig. 12.19), y el manejo de propiedades multivalor requiere especial cuidado. Además, aunque las consultas en estrella se pueden gestionar eficientemente, este enfoque podría no ser muy útil con otros tipos de consultas. Finalmente, cuando se utiliza la asignación manual, la agrupación de propiedades "similares" no es trivial y las malas decisiones de diseño agravan el problema del valor nulo.

Tablas binarias

El enfoque de tablas binarias sigue la organización de esquemas de bases de datos orientadas a columnas y define una tabla de dos columnas para cada propiedad que contiene el sujeto y el objeto. Esto da como resultado un conjunto de tablas, cada una ordenada por sujeto. Esta es una organización típica de bases de datos orientadas a columnas y se beneficia de las ventajas habituales de estos sistemas, como la reducción de E/S al leer solo las propiedades necesarias y la longitud reducida de las tuplas, la compresión gracias a la redundancia en los valores de las columnas, etc. Además, evita los valores nulos que se experimentan en las tablas de propiedades, así como la necesidad de algoritmos de agrupamiento manuales o automáticos para propiedades similares, y admite propiedades multivalor; cada una se convierte en una fila independiente, como en el caso de la tabla DS de Jena. Además, dado que las tablas se ordenan por sujetos, las uniones sujeto-sujeto se pueden implementar mediante uniones de fusión eficientes. Las desventajas son que las consultas requieren más operaciones de unión, algunas de las cuales pueden ser uniones sujeto-objeto que no se ven facilitadas por la operación de unión de fusión. Además, las inserciones en las tablas tienen una mayor sobrecarga, ya que es necesario actualizar varias tablas. Se ha argumentado que el problema de la inserción puede mitigarse mediante inserciones por lotes, pero en repositorios RDF dinámicos, es probable que la dificultad de las inserciones siga siendo un problema significativo. La proliferación del número de tablas puede afectar negativamente la escalabilidad (en cuanto al número de propiedades) del enfoque de tablas binarias.

Ejemplo 12.18. Por ejemplo, la representación de tabla binaria del conjunto de datos del Ejemplo 12.14 generaría una tabla para cada propiedad única (hay 18). Dos de estas tablas se muestran en la Figura 12.21.

Procesamiento basado en gráficos

Los enfoques de procesamiento RDF basados en grafos implementan fundamentalmente la semántica de las consultas RDF, tal como se definió al principio de esta sección. En otras palabras, mantienen la estructura gráfica de los datos RDF (utilizando alguna representación como listas de adyacencia), convierten la consulta SPARQL en un grafo de consultas y realizan subgrafos.

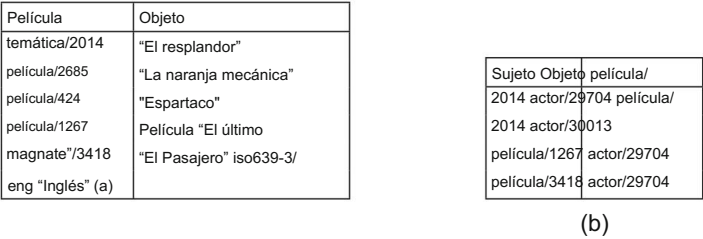


Fig. 12.21 Organización de la tabla binaria de las propiedades (a) "rdfs:label" y (b) "movie:actor" del conjunto de datos de ejemplo (se eliminan los prefijos)

coincidencia mediante homomorfismo para evaluar la consulta contra el gráfico RDF. Sistemas como gStore y chameleon-db siguen este enfoque.

La ventaja de este enfoque es que mantiene la representación original de los datos RDF y aplica la semántica prevista de SPARQL. La desventaja es el coste de la coincidencia de subgrafos: el homomorfismo de grafos es NP-completo. Esto plantea problemas con respecto a la escalabilidad de este enfoque a grafos RDF grandes; se pueden utilizar técnicas típicas de bases de datos, incluyendo la indexación, para abordar este problema. A continuación, presentamos el enfoque en el contexto del sistema gStore para destacar los

problemas. gStore utiliza la representación de grafos mediante listas de adyacencia. Codifica cada vértice de entidad y clase en una cadena de bits de longitud fija que captura la información de "vecindad" para cada vértice y la explota durante la coincidencia de grafos. Esto da como resultado la generación de un grafo de firma de datos  $G_s$ , en el que cada vértice corresponde a un vértice de clase o entidad en el grafo RDF original  $G$ , junto con las aristas cuyos extremos son vértices de entidad o clase. La Figura 12.22a muestra el grafo de firma de datos  $G_s$ , que corresponde al grafo RDF  $G$  de la Figura 12.16. Una consulta SPARQL entrante también se representa como un grafo de consulta  $Q$ , codificado

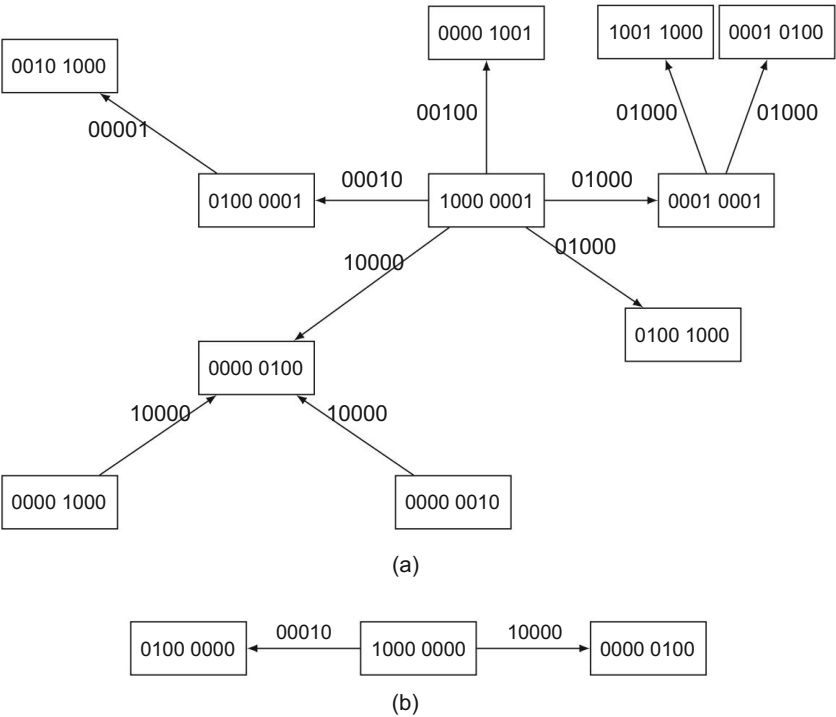


Fig. 12.22 Gráficos de firma. (a) Gráfico de firma de datos  $G_s$ . (b) Gráfico de firma de consulta  $Q_s$ .



Gráfico de firma  $Q$ . La codificación del gráfico de consulta representado en la Fig. 12.17 en un gráfico de firma de consulta  $Q_2$  se muestra en la Fig. 12.22b.

El problema ahora se convierte en encontrar coincidencias de  $Q$  sobre  $G$ . Aunque tanto el grafo RDF como el grafo de consulta son más pequeños como resultado de la codificación, la NP-completitud del problema persiste. Por lo tanto, gStore utiliza una estrategia de filtrado y evaluación para reducir el espacio de búsqueda sobre el que se aplica la coincidencia. El objetivo es, primero, utilizar una estrategia de poda de falsos positivos para encontrar un conjunto de subgrafos candidatos (denotados como CL) y, luego, validarlos utilizando la lista de adyacencia para encontrar respuestas (denotadas como RS). En consecuencia, se deben abordar dos cuestiones. En primer lugar, la técnica de codificación debe garantizar que  $RS \subseteq CL$ ; la codificación descrita anteriormente logra esto demostrablemente. En segundo lugar, se requiere un algoritmo de coincidencia de subgrafos eficiente para encontrar coincidencias de  $Q$  sobre  $G$ . Para esto, gStore utiliza una estructura de índice llamada VS-tree que es un grafo resumen de  $G$ . El árbol VS se utiliza para procesar consultas de manera eficiente utilizando una estrategia de poda para reducir el espacio de búsqueda para encontrar coincidencias de  $Q$  sobre  $G$ .

### Ejecución SPARQL distribuida y federada

A medida que crecen las colecciones RDF, se han desarrollado soluciones de escalamiento horizontal que implican procesamiento paralelo y distribuido. Muchas de estas soluciones dividen un grafo RDF  $G$  en varios fragmentos y ubican cada uno en un sitio diferente dentro de un sistema paralelo/distribuido. Cada sitio alberga un almacén RDF centralizado de algún tipo.

En tiempo de ejecución, una consulta SPARQL  $Q$  se descompone en varias subconsultas, de modo que cada subconsulta pueda responderse localmente en un sitio, y los resultados se agregan posteriormente. Cada uno de estos artículos propone su propia estrategia de partición de datos, y las diferentes estrategias de partición resultan en diferentes métodos de procesamiento de consultas. Algunos enfoques utilizan soluciones basadas en MapReduce, donde las tripletas RDF se almacenan en HDFS y cada patrón de tripleta se evalúa mediante el escaneo de los archivos HDFS, seguido de una implementación de unión de MapReduce. Otros enfoques siguen, en mayor o menor medida, las metodologías de procesamiento de consultas distribuidas/paralelas descritas en detalle en varios capítulos de este libro, donde la consulta se particiona en subconsultas y se evalúa en todos los sitios.

Una alternativa propuesta es utilizar la evaluación parcial de consultas para ejecutar consultas SPARQL distribuidas. La evaluación parcial de funciones es una estrategia bien conocida en lenguajes de programación cuya idea básica es la siguiente: dada una función  $f(s, d)$ , donde  $s$  es la entrada conocida y  $d$  es la entrada aún no disponible, la parte del cálculo de  $f$  que depende únicamente de  $s$  genera una respuesta parcial. En este enfoque, los datos se particionan, pero las consultas no: cada sitio recibe la consulta SPARQL completa  $Q$  y la ejecuta en el fragmento de grafo RDF local, lo que proporciona cálculo paralelo de datos.

En este contexto particular, la estrategia de evaluación parcial se aplica de la siguiente manera: cada sitio  $S_i$  considera el fragmento  $F_i$  como la entrada conocida en la etapa de evaluación parcial; la entrada no disponible es el resto del grafo ( $G = G \setminus F_i$ ). Hay dos cuestiones importantes que abordar en este marco. La primera es calcular los resultados de la evaluación parcial en cada sitio  $S_i$  dado un grafo de consulta  $Q$ ; en otras palabras, abordar el homomorfismo de grafos de  $Q$  sobre  $F_i$ ; esto se denomina coincidencia parcial local, ya que

Encuentra las coincidencias internas del fragmento Fi. Dado que no es posible garantizar la disjunción de aristas en la partición disjunta de vértices, habrá aristas cruzadas entre fragmentos de grafos. La segunda tarea consiste en ensamblar estas coincidencias parciales locales para calcular las coincidencias cruzadas. Esta tarea de ensamblaje puede ejecutarse en un sitio de control o de forma similar a una unión distribuida.

Los enfoques anteriores toman un conjunto de datos RDF centralizado y lo particionan para su ejecución distribuida/paralela. En muchos entornos RDF, surgen problemas similares a los que abordamos en la integración de bases de datos, que requieren una solución federada. En el entorno RDF, algunos sitios que alojan datos RDF también pueden procesar consultas SPARQL; estas se denominan puntos finales SPARQL. Un ejemplo típico es LOD, donde diferentes repositorios RDF se interconectan, lo que proporciona una base de datos distribuida virtualmente integrada. Una técnica común en entornos RDF federados consiste en precalcular metadatos para cada punto final SPARQL. Los metadatos pueden especificar las capacidades del punto final, una descripción de los patrones triples (es decir, propiedades) que se pueden responder en ese punto final, u otra información que utiliza el algoritmo en cuestión. A partir de los metadatos, la consulta SPARQL original se descompone en varias subconsultas, cada una de las cuales se envía a sus puntos finales SPARQL correspondientes. Luego, los resultados de las subconsultas se unen para responder la consulta SPARQL original.

Una alternativa al precálculo de metadatos es utilizar consultas SPARQL ASK para recopilar información sobre cada punto final y construir los metadatos sobre la marcha. Con base en los resultados de estas consultas, una consulta SPARQL se descompone en subconsultas y se asigna a los puntos finales.

### 12.6.2.3 Navegación y consulta del LOD

LOD consiste en un conjunto de documentos web. Por lo tanto, el punto de partida es un documento web con tripletas RDF integradas que codifican recursos web. Las tripletas RDF contienen enlaces de datos a otros documentos que permiten interconectarlos para obtener la estructura gráfica.

La semántica de las consultas SPARQL sobre el LOD se vuelve compleja. Una posibilidad es adoptar una semántica web completa que especifique que el alcance de la evaluación de una expresión de consulta SPARQL abarca todos los datos vinculados. No se conoce ningún algoritmo de ejecución de consultas (de terminación) que garantice la completitud de los resultados bajo esta semántica. La alternativa es una familia de semánticas basadas en la accesibilidad que definen el alcance de la evaluación de una consulta SPARQL en función de los documentos accesibles: dado un conjunto de URIs de semilla y una condición de accesibilidad, el alcance abarca todos los datos a lo largo de las rutas de los enlaces de datos desde las semillas y que satisfacen la condición de accesibilidad. La familia se define mediante diferentes condiciones de accesibilidad. En este caso, existen algoritmos computacionalmente viables.

Existen tres enfoques para la ejecución de consultas SPARQL sobre LOD: basado en recorrido, basado en índice e híbrido. Los enfoques de recorrido implementan básicamente una semántica basada en la accesibilidad: a partir de las URI de origen, descubren recursivamente las URI relevantes recorriendo enlaces de datos específicos durante la ejecución de la consulta. Para estos...

En los algoritmos, la selección de las URI de origen es crucial para el rendimiento. La ventaja de los enfoques transversales es su simplicidad de implementación, ya que no requieren el mantenimiento de estructuras de datos (como índices). Las desventajas son la latencia en la ejecución de consultas, ya que estos algoritmos "exploran" documentos web, y la recuperación repetida de datos de cada documento introduce una latencia significativa. Además, presentan una posibilidad limitada de paralelización; se pueden paralelizar en la misma medida que los algoritmos de rastreo.

Los enfoques basados en índices utilizan un índice para determinar las URI relevantes, lo que reduce el número de documentos vinculados a los que se debe acceder. Una clave de índice razonable son los patrones triples, en cuyo caso las URI relevantes para una consulta dada se determinan accediendo al índice, y la consulta se evalúa sobre los datos recuperados al acceder a dichas URI. En estos enfoques, la recuperación de datos puede paralelizarse completamente, lo que reduce el impacto negativo de la recuperación de datos en el tiempo de ejecución de la consulta. Las desventajas de este enfoque son la dependencia del índice —tanto en términos de la latencia que introduce su construcción como de la restricción que impone sobre lo que se puede seleccionar— y los problemas de actualización derivados del dinamismo de la web y la dificultad de mantener el índice actualizado.

Los enfoques híbridos realizan una ejecución transversal mediante un listado priorizado de URI para la búsqueda. Las semillas iniciales provienen de un índice precargado; las nuevas URI descubiertas que no están en el índice se clasifican según el número de documentos de referencia.

### 12.6.3 Problemas de calidad de los datos en la integración de datos web

En el capítulo 7 (específicamente en la sección 7.1.5) analizamos los problemas de calidad y limpieza de datos en sistemas de integración de bases de datos (principalmente almacenes de datos). Los problemas de calidad de datos en datos web se agravan debido a la gran cantidad de fuentes de datos web, el proceso de entrada de datos incontrolado en las fuentes de información web y la mayor diversidad de datos. La calidad de datos abarca tanto la consistencia como la veracidad de los datos (autenticidad y conformidad con la realidad). En un almacén de datos, la consistencia de los datos se obtiene mediante la limpieza de datos, que consiste en detectar y eliminar errores e inconsistencias. La limpieza de datos en el contexto web (y también en lagos de datos) se dificulta por la falta de información del esquema y las limitaciones de restricciones de integridad que se pueden definir sin un esquema.

Comprobar la veracidad de los datos sigue siendo un gran desafío. Sin embargo, si se superponen muchas fuentes de datos diferentes, como suele ocurrir con los datos procedentes de la web, por ejemplo, habrá un alto nivel de redundancia. Es posible utilizar técnicas eficientes de fusión de datos (que se analizarán en breve) para detectar los valores correctos de los mismos datos y, por lo tanto, descubrir la verdad.

En esta sección, destacamos algunos de los principales aspectos de calidad y limpieza de datos. problemas en los datos web y discutir las soluciones actuales para abordarlos.

12.6.3.1 Limpieza de datos web estructurados

Los datos estructurados representan una categoría importante de datos en la web y presentan numerosos problemas de calidad. A continuación, resumimos las técnicas propuestas para la limpieza de datos estructurados en general. Posteriormente, señalamos los desafíos específicos que presenta la limpieza de datos estructurados en la web.

La Figura 12.23 muestra un flujo de trabajo típico para la limpieza de datos estructurados, que consta de un paso opcional de descubrimiento y perfilado, un paso de detección de errores y un paso de reparación de errores. Para limpiar un conjunto de datos corruptos, a menudo necesitamos modelar diversos aspectos de estos datos (metadatos), por ejemplo, esquemas, patrones, distribuciones de probabilidad y otros metadatos. Una forma de obtener dichos metadatos es consultando a expertos en la materia, lo cual suele ser un proceso costoso y lento. Por lo tanto, se suele utilizar un paso de descubrimiento y perfilado para descubrir estos metadatos automáticamente. Dado un conjunto de datos corruptos y los metadatos asociados, el paso de detección de errores encuentra parte de los datos que no se ajusta a los metadatos y declara que este subconjunto contiene errores. Los errores detectados por el paso de detección de errores pueden presentarse en diversas formas, como valores atípicos, violaciones de las restricciones de integridad y duplicados. Finalmente, el paso de reparación de errores produce actualizaciones de datos que se aplican al conjunto de datos corruptos para eliminar los errores detectados. Dado que existen muchas incertidumbres en el proceso de limpieza de datos, siempre que sea posible se consultan fuentes externas, como bases de conocimiento y expertos humanos, para garantizar la precisión del flujo de trabajo de limpieza.

El proceso anterior funciona bien para tablas estructuradas con un amplio conjunto de metadatos, por ejemplo, un esquema extenso con suficientes restricciones para modelar las interacciones entre columnas y filas. Además, el proceso de limpieza y detección de errores funciona mejor cuando hay suficientes ejemplos (tuplas) para que los algoritmos automáticos comparen varias instancias y detecten posibles errores, aprovechando la redundancia de los datos para corregirlos.

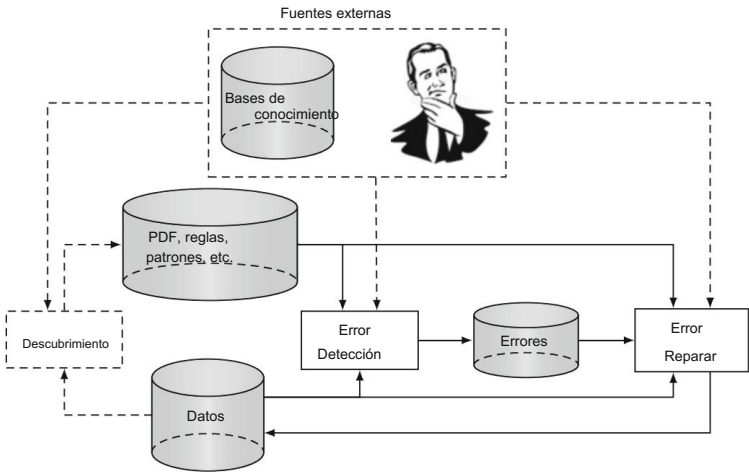


Fig. 12.23 Un flujo de trabajo típico para limpiar datos estructurados

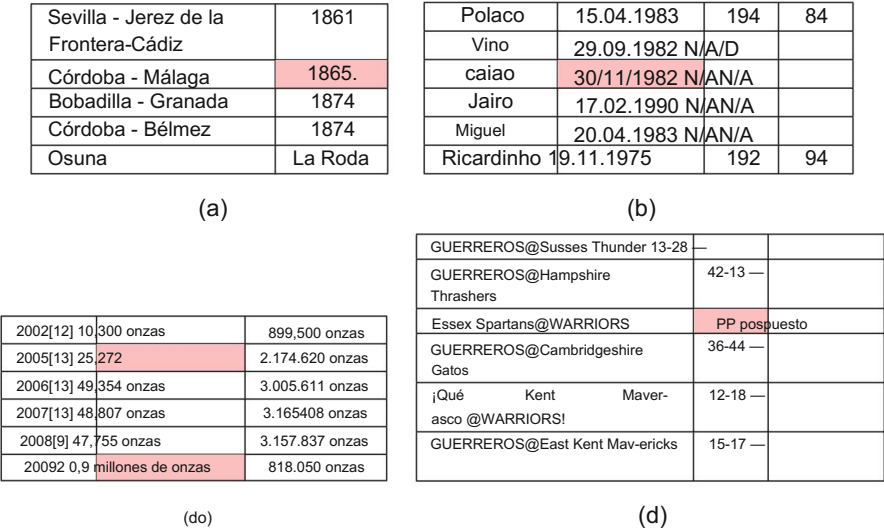


Fig. 12.24 Problemas de calidad de datos en datos web estructurados (los datos erróneos están marcados en celdas rojas). Adaptado de [Huang y He 2018]. (a) Punto extra. (b) Fechas mixtas. (c) Pesos inconsistentes. (d)

Markador de posición de puntuación

errores. Sin embargo, en las tablas web, ambas premisas no se cumplen, ya que la mayoría de Las tablas son cortas (pocas tuplas) y delgadas (número limitado de atributos). o hacer Para colmo, el número de tablas web es mucho mayor que el número de tablas en una almacén de datos. Esto significa que la limpieza manual, aunque relativamente para una sola web tabla, no es viable para todas las tablas web estructuradas. La Figura 12.24 muestra algunos ejemplos Se encontraron errores en las tablas de Wikipedia y se estima que hay unos 300 000 errores de este tipo.

12.6.3.2 Fusión de datos web

Un problema común que surge a menudo en la integración de datos web es la fusión de datos, es decir Decidir cuál es el valor correcto para un artículo que tiene diferentes representaciones de múltiples fuentes web. El problema es que diferentes fuentes web pueden proporcionar Representaciones contradictorias, lo que dificulta la fusión de datos. Hay dos tipos de conflictos de datos: incertidumbre y contradicción. La incertidumbre es un conflicto entre una valor no nulo y uno o más valores nulos que se utilizan para describir la misma propiedad de una entidad del mundo real. La incertidumbre se debe a la falta de información, generalmente representado por valores nulos en una fuente. La contradicción es un conflicto entre dos o más valores no nulos diferentes que representan diferentes valores de la misma propiedad de una entidad del mundo real. La contradicción se debe a que diferentes fuentes proporcionan información diferente. valores para el mismo atributo.

Por lo tanto, la limpieza automática de las tablas web es particularmente compleja. Durante la limpieza Las técnicas desarrolladas en el contexto del almacén de datos se pueden aplicar para limpiar algunos

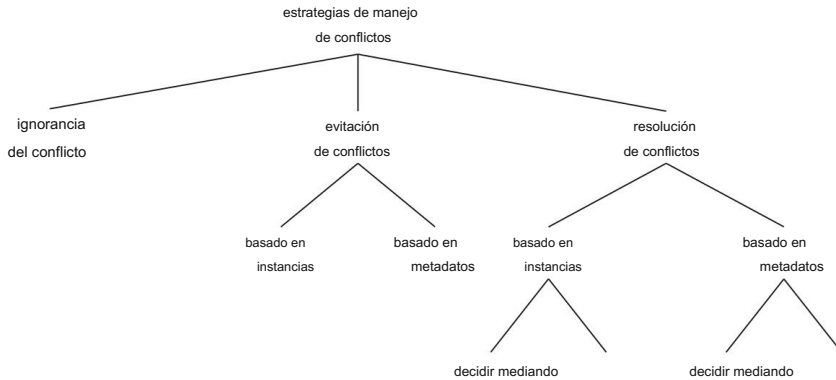


Figura 12.25 Clasificación de estrategias para la fusión de datos. De [Bleiholder y Naumann, 2009]

Errores: la limpieza de tablas web requiere técnicas más especializadas. Auto-Detect es una propuesta reciente que busca detectar dichos errores en tablas web. Auto-Detect es una técnica basada en estadísticas basadas en datos que aprovecha las estadísticas de coocurrencia de valores de grandes corpus para la detección de errores. La premisa principal es que si una determinada combinación de valores es extremadamente rara (cuantificada mediante información mutua puntual), sugiere un posible error. Si bien Auto-Detect puede detectar muchos errores, no sugiere soluciones para los datos. Aún no hemos visto propuestas que reparen automáticamente los errores (o incluso sugieran soluciones) en tablas web.

La Figura 12.25 muestra la clasificación de diferentes estrategias de fusión de datos. Las estrategias de ignorancia de conflictos ignoran los conflictos y simplemente los pasan a los usuarios o aplicaciones. Las estrategias de evitación de conflictos reconocen la existencia de representaciones conflictivas y aplican una regla simple para tomar una decisión única basada en la instancia de datos o los metadatos. Un ejemplo de estrategia de evitación de conflictos basada en instancias es preferir valores no nulos sobre valores nulos. Un ejemplo de estrategia de evitación de conflictos basada en metadatos es preferir valores de una fuente sobre valores de otra. Las estrategias de resolución de conflictos resuelven los conflictos eligiendo un valor de los valores ya presentes (decisión) o eligiendo un valor que no necesariamente exista entre los valores presentes (mediación). Un ejemplo de estrategia de resolución de conflictos de decisión basada en instancias es tomar el valor más frecuente. Un ejemplo de estrategia de resolución de conflictos de mediación basada en instancias es tomar el promedio de todos los valores presentes.

### 12.6.3.3 Calidad de la fuente web

Estas estrategias básicas de resolución de conflictos descritas anteriormente se basan principalmente en valores participativos para resolver conflictos, y pueden presentar deficiencias en los tres aspectos siguientes. En primer lugar, las fuentes web tienen cualidades diferentes; los valores de los datos proporcionados por fuentes web más precisas suelen ser más precisos. Sin embargo, las fuentes web más precisas...

Las fuentes también pueden proporcionar valores incorrectos; por lo tanto, a menudo se requiere una estrategia de resolución avanzada para considerar la calidad de la fuente al predecir el valor correcto. En segundo lugar, las fuentes web pueden copiarse entre sí, e ignorar este tipo de dependencias entre ellas puede generar decisiones de resolución erróneas. Por ejemplo, la estrategia de voto mayoritario para resolver conflictos se vería afectada si se copian algunos datos de una fuente. En tercer lugar, el valor correcto de un dato también puede evolucionar con el tiempo (p. ej., la afiliación de una persona); por lo tanto, es crucial distinguir entre un valor incorrecto y un valor obsoleto al evaluar la precisión de las fuentes y tomar decisiones de resolución.

El componente fundamental de las estrategias avanzadas de fusión de datos es evaluar la fiabilidad o calidad de una fuente. En esta sección, analizamos cómo se modela la precisión de una fuente de datos y mencionamos cómo se extiende dicho modelo para gestionar las dependencias y la actualización de las fuentes.

### Precisión de la fuente

La precisión de una fuente se mide como la fracción de valores verdaderos que proporciona. La precisión de una fuente  $S$  se denota por  $A(S)$ , que puede considerarse como la probabilidad de que un valor proporcionado por  $S$  sea el verdadero. Sea  $V(S)$  el valor proporcionado por  $S$ . Para cada  $v \in V(S)$ , sea  $P_r(v)$  la probabilidad de que  $v$  sea el verdadero. Entonces,  $A(S)$  se calcula de la siguiente manera:

$$A(S) = \frac{\sum_{v \in V(S)} P_r(v)}{|V(S)|}$$

Consideremos un dato  $D$ . Sea  $\text{Dom}(D)$  el dominio de  $D$ , que incluye un valor verdadero y  $n$  valores falsos. Sea  $SD$  el conjunto de fuentes que proporcionan un valor para  $D$ , y  $SD(v)$  el conjunto de fuentes que proporcionan el valor  $v$  para  $D$ . Sea  $(D)$  la observación de qué valor proporciona cada  $S \in SD$  para  $D$ . La probabilidad  $P_r(v)$  se puede calcular de la siguiente manera:

$$P_r(v) = P_r(v \text{ es valor verdadero} | (D)) = P_r((D) | v \text{ es valor verdadero})$$

Suponga que las fuentes son independientes y que los  $n$  valores falsos tienen la misma probabilidad. Para que suceda,  $P_r((D) | v \text{ es el valor verdadero})$  se puede calcular de la siguiente manera:

$$P_r((D) | v \text{ es el valor verdadero}) = \frac{\prod_{S \in SD(v)} A(S)}{\prod_{S \in SD \setminus SD(v)} (1 - A(S))}$$

que puede reescribirse como

$$P_r((D) | v \text{ es el valor verdadero}) = \frac{\prod_{S \in SD(v)} A(S)}{\prod_{S \in SD \setminus SD(v)} (1 - A(S))} = \frac{\prod_{S \in SD(v)} A(S)}{\prod_{S \in SD} (1 - A(S))} \cdot \prod_{S \in SD} (1 - A(S))$$

Desde  $S \in SD$   $\frac{1-A(S)}{n}$  es el mismo para todos los valores, tenemos

$$P(r(D)|v \text{ es el valor verdadero}) = \frac{nA(S)}{1 - A(S)} \quad S \in SD(v)$$

En consecuencia, el recuento de votos de una fuente de datos  $S$  se define como:

$$C(S) = \ln \frac{nA(S)}{1 - A(S)}$$

El recuento de votos de un valor  $v$  se define como:

$$C(v) = \sum_{S \in SD(v)} C(S)$$

Intuitivamente, una fuente con un mayor número de votos es más precisa y un valor con un mayor número de votos tiene mayor probabilidad de ser verdadero. Combinando el análisis anterior, la probabilidad de cada valor  $v$  se puede calcular de la siguiente manera:

$$P(r(v)) = \frac{\exp(C(v))}{\sum_{v \in \text{inDom}(v)} \exp(C(v))}$$

Obviamente, para un dato  $D$ , el valor  $v \in \text{Dom}(D)$  con la mayor probabilidad  $P(r(v))$  se seleccionaría como el valor verdadero. Como podemos ver, el cálculo de la precisión de la fuente  $A(S)$  depende de la probabilidad  $P(r(v))$ , y el cálculo de la probabilidad  $P(r(v))$  depende de la precisión de la fuente  $A(S)$ . Es posible un algoritmo que parte de la misma precisión para cada fuente y la misma probabilidad para cada valor, y calcula iterativamente las probabilidades para todas las fuentes y para todos los valores hasta la convergencia. El criterio de convergencia se establece cuando no hay cambios en las precisiones de la fuente ni oscilaciones en los valores verdaderos determinados.

#### Dependencia de la fuente

El cálculo anterior para la precisión de la fuente supone que las fuentes son independientes. En realidad, las fuentes se copian entre sí, lo que crea dependencias. Existen dos supuestos para la detección de copias entre fuentes. En primer lugar, para un dato en particular, solo existe un valor verdadero, pero suele haber múltiples valores falsos. Que dos fuentes compartan el mismo valor verdadero no implica necesariamente dependencia; sin embargo, que dos fuentes compartan el mismo valor falso suele ser poco frecuente y, por lo tanto, es más probable que implique dependencia entre ellas. En segundo lugar, un subconjunto aleatorio de valores proporcionado por una fuente de datos suele tener una precisión similar a la del conjunto completo de valores proporcionados por la fuente de datos. Sin embargo, en una fuente de datos copiadora, el subconjunto de valores que copia puede tener una precisión diferente a la del resto de los valores que proporciona de forma independiente.



Por lo tanto, entre dos fuentes dependientes donde una copia a otra, es más probable que la fuente cuya precisión en los valores de sus datos difiera significativamente de los valores compartidos con la otra sea la copiadora. Con base en estas intuiciones, se puede desarrollar un modelo bayesiano para calcular la probabilidad de copia entre dos fuentes  $S_1$  y  $S_2$ , dadas las observaciones de todos los datos; esta probabilidad se utiliza para ajustar el cálculo del recuento de votos a un valor  $C(v)$  que considere las dependencias entre las fuentes.

### Frescura de la fuente

Hasta ahora hemos asumido que la fusión de datos se realiza en una instantánea estática de los datos. Sin embargo, en realidad, los datos evolucionan con el tiempo y el valor real de un artículo también puede cambiar. Por ejemplo, la hora de salida programada de un vuelo puede cambiar según el mes; la afiliación de una persona puede cambiar con el tiempo; y el director ejecutivo de una empresa también. Para capturar estos cambios, las fuentes de datos deben actualizar sus datos. En este entorno dinámico, los errores de datos se producen por diversas razones: (1) las fuentes pueden proporcionar valores erróneos, similar a la configuración estática; (2) las fuentes pueden no actualizar sus datos en absoluto; y (3) algunas fuentes pueden no actualizar sus datos a tiempo. La fusión de datos, en este contexto, busca encontrar todos los valores correctos y sus períodos válidos en el historial, cuando los valores reales evolucionan con el tiempo.

Si bien la calidad de la fuente se puede medir mediante la precisión en el caso estático, las métricas para evaluarla son más complejas en el entorno dinámico: una fuente de alta calidad debería proporcionar un nuevo valor para un dato si, y solo si, e inmediatamente después de que el valor se convierta en el verdadero. Se pueden utilizar tres métricas para comprender esta intuición: la cobertura de una fuente mide las transiciones de los diferentes datos que captura; la exactitud mide el porcentaje de transiciones que una fuente captura erróneamente (al proporcionar un valor erróneo); y la frescura mide la rapidez con la que una fuente captura un cambio de valor. De nuevo, es posible basarse en el análisis bayesiano para determinar tanto el tiempo como el valor de cada transición para un dato.

El aprendizaje automático y los modelos probabilísticos también se han utilizado en la fusión de datos y el modelado de la calidad de las fuentes de datos. En particular, SLiMFAST es un marco que expresa la fusión de datos como un problema de aprendizaje estadístico sobre modelos probabilísticos discriminativos. A diferencia de los enfoques de fusión basados en el aprendizaje previos, SLiMFAST ofrece garantías de calidad para los resultados fusionados y también puede incorporar el conocimiento del dominio disponible en el proceso de fusión. La Figura 12.26 muestra la descripción general del sistema SLiMFAST. La entrada a SLiMFAST incluye (1) un conjunto de observaciones de la fuente, concretamente los valores posiblemente conflictivos proporcionados para diferentes objetos por diferentes fuentes; (2) un conjunto opcional de datos de verdad etiquetados, concretamente los valores reales para un subconjunto de objetos; y (3) conocimiento del dominio sobre las fuentes que los usuarios consideran informativo sobre la precisión de las fuentes de datos. SLiMFAST recopila toda esta información en un modelo gráfico probabilístico para el aprendizaje y la inferencia holísticos. Dependiendo de la cantidad de datos de verdad de campo disponibles, SLiMFAST decidirá qué algoritmo (maximización de expectativas o minimización de pérdidas empíricas) utilizar para aprender los parámetros de los modelos gráficos.

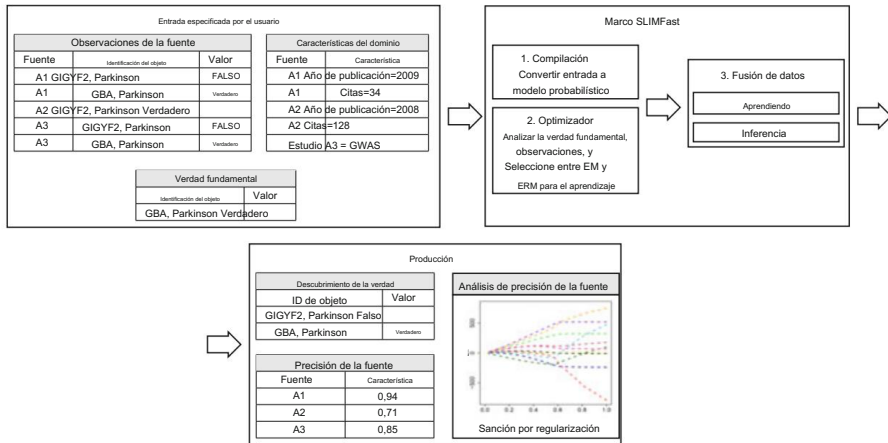


Fig. 12.26 Descripción general de SLIMFast. De [Rekatsinas et al., 2017]

El modelo aprendido se utiliza luego para inferir tanto el valor de los objetos como la fuente. precisiones, como se muestra en la salida.

## 12.7 Notas bibliográficas

Hay varias buenas fuentes sobre temas web, cada una con un enfoque ligeramente diferente. Enfoque. Abiteboul et al. [2011] se centran en el uso de XML y RDF para datos web. modelado y también contienen discusiones sobre búsqueda y tecnologías de big data como como MapReduce. Una perspectiva de almacenamiento de datos web se ofrece en [Bhowmick et al. 2004]. Bonato [2008] se centra principalmente en el modelado de la web como un gráfico. y cómo se puede explotar este gráfico. Trabajos iniciales sobre los lenguajes de consulta web y Los enfoques se analizan en [Abiteboul et al. 1999].

Una muy buena visión general de los problemas de búsqueda web es [Arasu et al. 2001], que también Sigue en la sección 12.2. Además, Lawrence y Giles [1998] proporcionan una explicación anterior. Discusión sobre el mismo tema, centrada en la web abierta. Florescu et al. [1998] encuesta Problemas de búsqueda web desde la perspectiva de una base de datos. La web profunda (oculta) es el tema de... [Raghavan y García-Molina 2001]. Lage et al. [2002] y Hedley et al. [2004b] También se analiza la búsqueda en la web profunda y el análisis de los resultados. Metabúsqueda. El acceso a la web profunda se analiza en [Ipeirotis y Gravano 2002, Callan y Connell 2001, Callan et al. 1999, Hedley et al. 2004a]. La metabúsqueda relacionada El problema de la selección de bases de datos es analizado por Ipeirotis y Gravano [2002] y Gravano et al. [1999] (algoritmo GIOSS).

Las estadísticas sobre la web abierta se toman de [Bharat y Broder 1998, Lawrence y Giles 1998, 1999, Gulli y Signorini 2005] y aquellos relacionados con la web profunda se deben a [Hirate et al. 2006] y [Bergman 2001].

La estructura gráfica de la web y el uso de gráficos para modelar y consultar la web es el tema de muchas publicaciones: [Kumar et al. 2000, Raghavan y Garcia-Molina 2003, Kleinberg et al. 1999] analizan el modelado de gráficos web, [Kleinberg et al. 1999, Brin y Page 1998, Kleinberg 1999] se centran en los gráficos para la búsqueda, y [Chakrabarti et al. 1998] para la categorización y clasificación del contenido web. La discusión sobre las características del gráfico web y su estructura de pajarita se deben a Bonato [2008], Broder et al. [2000] y Kumar et al. [2000]. No discutimos los problemas importantes relacionados con la gestión del gráfico web muy grande, dinámico y volátil. Estos están más allá del alcance de este capítulo, pero se pueden identificar dos líneas de investigación. El primero comprime el gráfico web para un almacenamiento y manipulación más eficientes [Adler y Mitzenmacher 2001], mientras que el segundo sugiere una representación especial para el gráfico web llamada S-nodos [Raghavan y García-Molina 2003].

Las cuestiones sobre el rastreo web son el tema de [Cho et al. 1998, Najork y Wiener 2001] y [Page et al. 1998], siendo este último el artículo clásico sobre PageRank cuya forma revisada, como se analiza en este capítulo, se debe a Langville y Meyer [2006]. Enfoques alternativos de rastreo son objeto de estudio en [Cho y García-Molina 2000] (basado en la frecuencia de cambio), [Cho y Ntoulas 2002] (basado en el muestreo) y [Edwards et al. 2001] (incremental). Las técnicas de clasificación para evaluar la relevancia se analizan en [Mitchell 1997, Chakrabarti et al. 2002] (Bayes ingenuo), Passerini et al. [2001], Altingövdé y Ulusoy [2004] (extensiones de Bayesian), y por McCallum et al. [1999], Kaelbling et al. [1996] (aprendizaje de refuerzo).

La indexación web es un tema importante que abordamos en la sección 12.2.2. Diversos métodos de indexación de texto se describen en [Manber y Myers, 1990] (matrices de sufijos), [Hersh, 2001] [Lim et al., 2003] (índices invertidos) y [Faloutsos y Christodoulakis, 1984] (archivos de firmas). Salton [1989] es probablemente la fuente clásica para el procesamiento y análisis de texto. Arasu et al. [2001], Melnik et al. [2001] y Ribeiro-Neto y Barbosa [1998] abordan los desafíos de crear índices invertidos para la web y sus soluciones. En relación con esto, la clasificación ha sido objeto de una amplia investigación. Además del conocido PageRank, el algoritmo HITS se debe a Kleinberg [1999].

En nuestro análisis del enfoque de datos semiestructurados para consultas web, destacamos el modelo de datos OEM y el lenguaje Lorel para exponer los conceptos. Estos se discuten en [Papakonstantinou et al. 1995] y [Abiteboul et al. 1997]. Las guías de datos para simplificar el OEM se discuten en [Goldman y Widom 1997]. UnQL [Buneman et al. 1996] presenta conceptos similares a Lorel. Nuestro análisis de lenguajes de consulta web en la Sección 12.3.2 dividió los lenguajes en primera y segunda generación; esto se debe a Florescu et al. [1998]. Los lenguajes de primera generación incluyen WebSQL [Mendelzon et al. 1997], W3QL [Konopnicki y Shmueli 1995] y WebLog [Lakshmanan et al. 1996]. Los lenguajes de segunda generación incluyen WebOQL [Arocena y Mendelzon, 1998] y StruQL [Fernández et al., 1997]. En el enfoque de consulta-respuesta, nos referimos a varios sistemas: Mulder [Kwok et al., 2001], WebQA [Lam y Özsu, 2002], Start [Katz y Lin, 2002] y Tritus [Agichtein et al., 2004].

Antoniou y Plexousakis [2018] presentan los componentes de la web semántica . Bizer et al. [2018] y Berners-Lee [2006] analizan la visión de los Datos Abiertos Vinculados (LOD) y sus requisitos. [Schmachtenberg et al., 2014] describe la separación de los dominios temáticos de los LOD .

Nuestra discusión sobre RDF se basa principalmente en [Özsu 2016]. Se describen cinco enfoques principales para gestionar datos RDF: (1) mapeo relacional directo—Angles y Gutierrez [2008], Sequeda et al. [2014] discuten el mapeo de SPARQL a SQL, Broekstra et al. [2002], y Chong et al. [2005] discuten Sesame SQL92SAIL y Oracle, respectivamente; (2) usando una sola tabla con indexación extensa (Hexastore [Weiss et al. 2008] y RDF-3X [Neumann y Weikum 2008, 2009]); (3) tablas de propiedades (Jena [Wilkinson 2006]; DB2RDF de IBM [Bornea et al. 2013]); (4) tablas binarias (SW-Store [Abadi et al. 2009], basada en la propuesta de Abadi et al. [2007]) , cuyos problemas de proliferación de tablas se analizan en [Sidiourgos et al. 2008]; (5) basadas en grafos ([Bönström et al. 2003], gStore [Zou et al. 2011, 2014] y chameleon-db [Aluç 2015]). Zou y Özsu [2017] analizan en detalle las técnicas basadas en grafos . La ejecución distribuida y basada en la nube de SPARQL se aborda en [Kaoudi y Manolescu 2015]. Se identifican tres enfoques para la ejecución de consultas SPARQL sobre LOD [Hartig 2013a]: basado en recorrido [Hartig 2013b, Ladwig y Tran 2011], basado en índice [Umbrich et al. 2011] e híbrido [Ladwig y Tran 2010].

La limpieza de datos estructurados se ha estudiado extensamente en entornos de integración de almacenes [Rahm y Do, 2000] e [Ilyas y Chu, 2015]. La expansión a un contexto más amplio, incluida la web, se aborda en Ilyas y Chu [2019]. En nuestro análisis de la fusión de datos (Sección 12.6.3.2), la separación de los conflictos de datos en incertidumbre y contradicción se debe a Dong y Naumann [2009]. En la misma sección, el análisis de Autodetección se debe a [Huang y He, 2018] y el análisis de la clasificación (así como la Fig. 12.25) es de [Bleiholder y Naumann, 2009]. El análisis de la precisión del modelado de la fuente de datos en la Sección 12.6.3.3, su extensión para gestionar las dependencias de la fuente y la frescura de la fuente se deben a Dong et al. [2009b,a].

Para un análisis más completo de la fusión avanzada de datos, se recomienda consultar el tutorial [Dong y Naumann, 2009] y el libro [Dong y Srivastava, 2015]. El sistema SlimFAST (véase la figura 12.26) se presenta en [Rekatsinas et al., 2017] y [Koller y Friedman, 2009].

Uno de los primeros sistemas en abordar problemas de limpieza de datos en lagos de datos es CLAMS [Farid et al., 2016], que permite descubrir y aplicar restricciones de integridad sobre los datos de un lago de datos. CLAMS utiliza un modelo de datos gráfico, basado en RDF, y un nuevo formalismo de restricciones de integridad para capturar tanto las restricciones relacionales como las reglas de calidad más expresivas basadas en patrones de grafos como restricciones de negación [Chu et al. 2013]. CLAMS también utiliza Spark y algoritmos paralelos para aplicar las restricciones y detectar inconsistencias en los datos.

## Ceremonias

Problema 12.1 ¿En qué se diferencia la búsqueda web de la consulta web?

Problema 12.2 (\*\*) Considere la arquitectura del motor de búsqueda genérico en la Figura 12.2. Proponga una arquitectura para un sitio web con un clúster sin recursos compartidos que implemente todos los componentes de esta figura, así como servidores web, en un entorno que admita grandes conjuntos de documentos web, índices muy grandes y un gran número de usuarios. Defina cómo se deben particionar y replicar las páginas web en el directorio de páginas y los índices. Analice las principales ventajas de su arquitectura en cuanto a escalabilidad, tolerancia a fallos y rendimiento.

Problema 12.3 (\*\*) Considere su solución del Problema 12.2. Ahora considere una consulta de búsqueda de palabras clave de un cliente web al motor de búsqueda. Proponga una estrategia de ejecución paralela para la consulta que clasifica las páginas web de resultados, con un resumen de cada página.

Problema 12.4 (\*) Para aumentar la proximidad del acceso y el rendimiento en diferentes regiones geográficas, proponga una extensión de la arquitectura del sitio web del Problema 12.3 con múltiples sitios, donde las páginas web se replican en todos ellos. Defina cómo se replican las páginas web. Defina también cómo se enruta una consulta de usuario a un sitio web. Analice las ventajas de su arquitectura en cuanto a escalabilidad, disponibilidad y rendimiento.

Problema 12.5 (\*) Considere su solución del Problema 12.4. Ahora considere una consulta de búsqueda de palabras clave de un cliente web al motor de búsqueda. Proponga una estrategia de ejecución paralela para la consulta que clasifica las páginas web de resultados, con un resumen de cada página.

Problema 12.6 (\*\*) Considere dos fuentes de datos web que modelamos como relaciones EMP1(Nombre, Ciudad, Teléfono) y EMP2(Nombre, Apellido, Ciudad). Tras la integración de esquemas, suponga que la vista EMP(Nombre, Nombre, Ciudad, Teléfono) está definida sobre EMP1 y EMP2, donde cada atributo de EMP proviene de un atributo de EMP1 o EMP2, y que EMP2 y Apellido se renombran como Nombre. Analice las limitaciones de dicha integración. Ahora, suponga que las dos fuentes de datos web son XML. Proporcione una definición correspondiente de los esquemas XML de EMP1 y EMP2. Proponga un esquema XML que integre EMP1 y EMP2 y evite los problemas identificados con EMP.

## Apéndice A

### Descripción general de los DBMS relacionales

Consulte <https://cs.uwaterloo.ca/ddbs>.

## Apéndice B

### Procesamiento centralizado de consultas

Consulte <https://cs.uwaterloo.ca/ddbs>.

## Apéndice C

### Fundamentos del procesamiento de transacciones

Consulte <https://cs.uwaterloo.ca/ddbs>.



## Apéndice D

### Revisión de redes de computadoras

Consulte <https://cs.uwaterloo.ca/ddbs>.

# Referencias

- Abadi, DJ, Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N. y Zdonik, S. (2003). Aurora: un nuevo modelo y arquitectura para la gestión de flujos de datos. VLDB J., 12(2):120–139.
- Abadi, DJ, Ahmad, Y., Balazinska, M., Çetintemel, U., Cherniack, M., Hwang, J.-H., Lindner, W., Maskey, A., Rasin, A., Ryvkina, E., Tatbul, N., Xing, Y., y Zdonik, SB (2005). El diseño del motor de procesamiento de flujos Borealis. En Actas de la 2.ª Conferencia Bienal sobre Investigación en Sistemas de Datos Innovadores, págs. 277–289.
- Abadi, DJ, Marcus, A., Madden, SR y Hollenbach, K. (2007). Gestión escalable de datos web semánticos mediante particionamiento vertical. En Actas de la 33.ª Conferencia Internacional sobre Bases de Datos Muy Grandes, páginas 411–422.
- Abadi, DJ, Marcus, A., Madden, S. y Hollenbach, K. (2009). SW-Store: un SGBD con particiones verticales para la gestión de datos de la web semántica. VLDB J., 18(2):385–406.
- Aberer, K. (2001). P-grid: Una estructura de acceso autoorganizada para sistemas de información P2P. En Proc. Int. Conf. sobre Sistemas de Inf. Cooperativa, páginas 179–194.
- Aberer, K. (2003). Introducción del editor invitado. Rec. ACM SIGMOD, 32(3):21–22.
- Aberer, K., Cudré-Mauroux, P., Datta, A., Despotovic, Z., Hauswirth, M., Ponceva, M. y Schmidt, R. (2003a). P-grid: un sistema P2P estructurado y autoorganizado. ACM SIGMOD Rec., 32 (3):29–33.
- Aberer, K., Cudré-Mauroux, P., y Hauswirth, M. (2003b). Empieza a tener sentido: La web habladora. Enfoque para acuerdos semánticos globales. J. Web Semantics, 1(1):89–114.
- Abiteboul, S., Quass, D., McHugh, J., Widom, J. y Wiener, J. (1997). El lenguaje de consulta Lorel. para datos semiestructurados. Int. J. Digit. Libr., 1(1):68–88.
- Abiteboul, S., Buneman, P. y Suciu, D. (1999). Datos en la Web: De las relaciones a Datos semiestructurados y XML. Morgan Kaufmann.
- Abiteboul, S., Manolescu, I., Rigaux, P., Rousset, M.-C., y Senellart, P. (2011). Gestión de datos web. Cambridge University Press.
- Abou-Rjeili, A. y Karypis, G. (2006). Algoritmos multinivel para particionar grafos de ley de potencia. En Proc. 20th IEEE Int. Parallel & Distributed Processing Symp., páginas 124–124.
- Abouzeid, A., Bajda-Pawlikowski, K., Abadi, D., Silberschatz, A. y Rasin, A. (2009). HadoopDB: una arquitectura híbrida de MapReduce y tecnologías DBMS para cargas de trabajo analíticas. Proc. VLDB Endowment, 2(1):922–933.
- Adali, S., Candan, KS, Papakonstantinou, Y. y Subrahmanian, V.S. (1996a). Caché de consultas y optimización en sistemas de mediación distribuidos. En Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos, páginas 137-148.

- Adali, S., Candan, KS, Papakonstantinou, Y. y Subrahmanian, V.S. (1996b). Caché de consultas y optimización en sistemas de mediación distribuidos. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*, páginas 137-148.
- Adamic, L. y Huberman, B. (2000). La naturaleza de los mercados en la World Wide Web. *Quart. J. Electron. Comm.*, 1:5-12.
- Adiba, M. (1981). Relaciones derivadas: Un mecanismo unificado para vistas, instantáneas y distribuidas. datos. En *Actas de la 7ª Conferencia Internacional sobre Bases de Datos Very*, páginas 293-305.
- Adiba, M. y Lindsay, B. (1980). Instantáneas de bases de datos. En *Actas de la 6.ª Conferencia Internacional sobre Bases de Datos Very*, páginas 86-91.
- Adler, M. y Mitzenmacher, M. (2001). Hacia la compresión de grafos web. En *Proc. Data. Conferencia sobre compresión*, páginas 203-212.
- Aggarwal, CC, editor. (2007). *Flujos de datos: modelos y algoritmos*. Springer.
- Agichtein, E., Lawrence, S. y Gravano, L. (2004). Aprender a encontrar respuestas a preguntas sobre la web. *ACM Trans. Internet Tech.*, 4(3):129-162.
- Agrawal, D. y Sengupta, S. (1993). Sincronización modular en bases de datos distribuidas y multiversión: Control de versiones y concurrencia. *IEEE Trans. Knowl. and Data Eng.*, 5 (1):126-137.
- Agrawal, D., Das, S., y El Abbadi, A. (2012). *Gestión de datos en la nube: Desafíos y oportunidades*. Conferencias de síntesis sobre gestión de datos. Morgan & Claypool Publishers.
- Agrawal, S., Narasayya, V. y Yang, B. (2004). Integración de particiones verticales y horizontales en el diseño automatizado de bases de datos físicas. En *Actas de la Conferencia Internacional ACM SIGMOD sobre Gestión de Datos*.
- Akal, F., Böhm, K. y Schek, H.-J. (2002). Evaluación de consultas OLAP en un clúster de bases de datos: Un estudio de rendimiento sobre paralelismo intraconsulta. En *Proc. 6.ª Conferencia de Europa del Este. Avances en Bases de Datos y Sistemas de Información*, páginas 218-231.
- Akal, F., Türker, C., Schek, H.-J., Breitbart, Y., Grabs, T. y Veen, L. (2005). Replicación y programación de grano fino con garantías de frescura y exactitud. En *Actas de la 31.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 565-576.
- Akbarinia, R. y Martins, V. (2007). Gestión de datos en el sistema APPA. *J. Grid Comp.*, 5 (3):303-317.
- Akbarinia, R., Martins, V., Pacitti, E. y Valduriez, P. (2006). Diseño e implementación de la arquitectura P2P de Atlas. En Baldoni, R., Cortese, G. y Davide, F., editores, *Global Data Management*, páginas 98-123. IOS Press.
- Akbarinia, R., Pacitti, E. y Valduriez, P. (2007a). Procesamiento de consultas top-k en hash distribuido. tablas. En *Proc. 13th Int. Euro-Par Conf.*, páginas 489-502.
- Akbarinia, R., Pacitti, E. y Valduriez, P. (2007b). Procesamiento de consultas en sistemas P2P. Técnico Informe 6112, INRIA, Rennes, Francia.
- Akbarinia, R., Pacitti, E. y Valduriez, P. (2007c). Algoritmos de mejor posición para consultas top-k. En *Actas de la 33.ª Conferencia Internacional sobre bases de datos muy grandes*, páginas 495-506.
- Akbarinia, R., Pacitti, E., y Valduriez, P. (2007d). Vigencia de datos en DHT replicados. En *Proc. Conferencia internacional ACM SIGMOD sobre gestión de datos*, páginas 211-222.
- Akida, T., Balikov, A., Bekiroglu, K., Chernyak, S., Haberman, J., Lax, R., McVeety, S., Mills, D., Nordstrom, P. y Whittle, S. (2013). MillWheel: Procesamiento de flujos con tolerancia a fallos a escala de internet. *Proc. VLDB Endowment*, 6(11):1033-1044.
- Alagiannis, I., Borovica, R., Branco, M., Idreos, S. y Ailamaki, A. (2012). NoDB: ejecución eficiente de consultas en archivos de datos sin procesar. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 241-252.
- Alagiannis, I., Idreos, S. y Ailamaki, A. (2014). H2O: Una tienda adaptativa manos libres. En *Proc. Conferencia internacional ACM SIGMOD sobre gestión de datos*, páginas 1103-1114.
- Alamoudi, AA, Grover, R., Carey, MJ y Borkar, VR (2015). Acceso a datos externos e indexación en AsterixDB. En *Actas de la 24.ª Conferencia Internacional de la ACM sobre Gestión de la Información y el Conocimiento*, páginas 3-12.
- Albutiu, M.-C., Kemper, A. y Neumann, T. (2012). Uniones masivas paralelas de ordenación y fusión en sistemas de bases de datos multinúcleo en memoria principal. *Proc. VLDB Endowment*, 5(10):1064-1075.

- Allard, T., Hébrail, G., Massegla, F. y Pacitti, E. (2015). Claroscuro: Transparencia y privacidad para la agrupación masiva de series temporales personales. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 779–794.
- Alomari, M., Cahill, M., Fekete, A. y Rohm, U. (2008). El coste de la serialización en plataformas que utilizan aislamiento de instantáneas. En *Actas de la 24.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 576–585.
- Alomari, M., Fekete, A. y Rohm, U. (2009). Una técnica robusta para garantizar ejecuciones serializables con un SGBD con aislamiento de instantáneas. En *Actas de la 25.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 341–352.
- Alsberg, PA y Day, JD (1976). Un principio para la compartición resiliente de recursos distribuidos. En *Actas de la 2.ª Conferencia Internacional sobre Ingeniería de Software*, páginas 562–570.
- Alsubaiee, S., Altowim, Y., Altwajiry, H., Behm, A., Borkar, VR, Bu, Y., Carey, MJ, Cetindil, I., Cheelangi, M., Faraaz, K., Gabrielova, E., Grover, R., Heilbron, Z., Kim, Y., Li, C., Li, G., Ok, JM, Onose, N., Pirzadeh, P., Tsostras, VJ, Vernica, R., Wen, J. y Westmann, T. (2014). AsterixDB: Un sistema de gestión de bases de datos escalable y de código abierto. *Proc. VLDB Endowment*, 7(14):1905–1916.
- Altingövdé, IS y Ulusoy, Ö. (2004). Explotación de reglas interclase para el rastreo enfocado. *IEEE Sistemas Inteligentes*, 19(6):66–73.
- Aluç, G. (2015). La carga de trabajo importa: Un enfoque robusto para el diseño físico de bases de datos RDF. Tesis doctoral, Universidad de Waterloo.
- Alvarez, V., Schuhknecht, FM, Dittrich, J. y Richter, S. (2014). Indexación adaptativa de memoria principal para sistemas multinúcleo. En *Actas del 10.º Taller sobre Gestión de Datos en Nuevo Hardware*, páginas 3:1–3:10.
- Amdahl, GM (1967). Validez del enfoque de procesador único para lograr capacidades de computación a gran escala. En *Proc. Spring Joint Computer Conf.*, páginas 483–485.
- Amsaleg, L., Franklin, MJ, Tomasic, A. y Urhan, T. (1996). Planes de consulta de cifrado para gestionar retrasos inesperados. En *Actas de la 4.ª Conferencia Internacional sobre Sistemas de Información Paralelos y Distribuidos*, páginas 208–219.
- Andreev, K. y Racke, H. (2006). Partición de grafos balanceados. *Theor. Comp. Sci.*, 39(6):929–939.
- Angles, R. y Gutiérrez, C. (2008). El poder expresivo de SPARQL. En *Proc. 7th Int. Semantic Web Conf.*, páginas 114–129.
- Antoniou, G. y Plexousakis, D. (2018). Web semántica. En Liu, L. y Özsu, MT, editores, *Enciclopedia de Sistemas de Bases de Datos*, páginas 3425–3429. Springer Nueva York, Nueva York, NY.
- Apache. (2016). Apache Giraph. <http://giraph.apache.org>. Último acceso junio de 2019.
- Apers, P., van den Berg, C., Flokstra, J., Grefen, P., Kersten, M. y Wilschut, A. (1992). Prisma/DB: un sistema de gestión de bases de datos relacional de memoria principal en paralelo. *IEEE Trans. Knowl. and Data Eng.*, 4:541–554.
- Apers, PMG (1981). Asignación redundante de relaciones en una red de comunicaciones. En *Actas del 5.º Taller de Berkeley sobre Gestión Distribuida de Datos y Redes de Computadores*, páginas 245–258.
- Arasu, A. y Widom, J. (2004). Semántica denotacional para consultas continuas sobre flujos y relaciones. *ACM SIGMOD Rec.*, 33(3):6–11.
- Arasu, A., Cho, J., García-Molina, H., Paepcke, A., y Raghavan, S. (2001). Búsqueda en la web. *ACM Trans. Tecnología de Internet*, 1(1):2–43.
- Arasu, A., Babu, S. y Widom, J. (2006). El lenguaje de consulta continua CQL: Fundamentos semánticos y ejecución de consultas. *VLDB J.*, 15(2):121–142.
- Armbrust, M., Xin, RS, Lian, C., Huai, Y., Liu, D., Bradley, J. K., Meng, X., Kaftan, T., Franklin, MJ, Ghodsi, A. y Zaharia, M. (2015). Spark SQL: Procesamiento de datos relacionales en Spark. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*, páginas 1383–1394.
- Arocena, G. y Mendelzon, A. (1998). WebOQL: Reestructuración de documentos, bases de datos y webs. En *Proc. 14ª Conf. Int. sobre Ingeniería de Datos*, páginas 24–33.
- Asad, O. y Kemme, B. (2016). Adaptcache: Particionamiento y migración de datos adaptativos para cachés de objetos distribuidos. En *Proc. ACM/IFIP/USENIX 17.ª Conferencia Internacional sobre Middleware*, páginas 7:1–7:13.
- Aspnes, J. y Shah, G. (2003). Grafos de salto. En *Proc. 14.º Simposio Anual ACM-SIAM sobre Algoritmos Discretos*, páginas 384–393.

- Avnur, R. y Hellerstein, J. (2000). Eddies: Procesamiento de consultas continuamente adaptativo. En Proc. Conferencia internacional ACM SIGMOD sobre gestión de datos, páginas 261–272.
- Ayad, A. y Naughton, J. (2004). Optimización estática de consultas conjuntivas con ventanas deslizantes sobre fuentes de información de flujo ilimitado. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 419–430.
- Azar, Y., Broder, AZ, Karlin, AR y Upfal, E. (1999). Asignaciones equilibradas. SIAM J. en Comput., 29(1):180–200.
- Babb, E. (1979). Implementación de una base de datos relacional mediante hardware especializado. ACM Trans. Sistema de bases de datos, 4(1):1–29.
- Babcock, B., Babu, S., Datar, M., Motwani, R. y Widom, J. (2002). Modelos y problemas en sistemas de flujo de datos. En Proc. Simposio ACM SIGACT-SIGMOD sobre Principios de Sistemas de Bases de Datos, páginas 1–16.
- Balazinska, M., Kwon, Y., Kuchta, N. y Lee, D. (2007). Moirae: Monitoreo mejorado por la historia. En Actas de la 3ª Conferencia Bienal sobre Investigación en Sistemas de Datos Innovadores, páginas 375–386.
- Balke, W.-T., Nejdl, W., Siberski, W. y Thaden, U. (2005). Recuperación progresiva distribuida de top-k en redes peer-to-peer. En Proc. 21.ª Conferencia Internacional sobre Ingeniería de Datos, páginas 174–185.
- Bancilhon, F. y Spyrratos, N. (1981). Actualización semántica de vistas relacionales. ACM Trans. Database Syst., 6(4):557–575.
- Barbara, D., Garcia-Molina, H., y Spauster, A. (1986). Políticas para la reasignación dinámica del voto. En Proc. 6ta Conferencia Internacional IEEE sobre Sistemas de Computación Distribuida, páginas 37–44.
- Barbara, D., Molina, HG y Spauster, A. (1989). Aumento de la disponibilidad bajo restricciones de exclusión mutua con reasignación dinámica del voto. ACM Trans. Comp. Syst., 7(4):394–426.
- Barthels, C., Loesing, S., Alonso, G. y Kossmann, D. (2015). Procesamiento de uniones en memoria a escala de rack mediante RDMA. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 1463–1475.
- Batini, C. y Lenzirini, M. (1984). Una metodología para la integración de esquemas de datos en entidades. modelo de relación. IEEE Trans. Softw. Eng., SE-10(6):650–654.
- Batini, C., Lenzirini, M. y Navathe, SB (1986). Un análisis comparativo de metodologías para Integración de esquemas de bases de datos. ACM Comput. Surv., 18(4):323–364.
- Beeri, C., Bernstein, PA, y Goodman, N. (1989). Un modelo de concurrencia en transacciones anidadas. sistemas. J. ACM, 36(2):230–269.
- Bell, D. y Grimson, J. (1992). Sistemas de bases de datos distribuidas. Addison Wesley. Reading.
- Bell, D. y Lapuda, L. (1976). Sistemas informáticos seguros: Exposición unificada e interpretación multics. Informe técnico MTR-2997 Rev.1, MITRE Corp, Bedford, MA.
- Berenson, H., Bernstein, P., Gray, J., Melton, J., O'Neil, E. y O'Neil, P. (1995). Una crítica de los niveles de aislamiento de ANSI SQL. En Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos, páginas 1–10.
- Bergamaschi, S. (2001). Integración semántica de fuentes de información heterogéneas. Data & Knowl. Eng., 36(3):215–249.
- Bergman, MK (2001). La web profunda: sacando a la luz el valor oculto. J. Electronic Publishing, 7(1).
- Bergsten, B., Couprie, M. y Valduriez, P. (1991). Prototipado de DBS3, un sistema de base de datos paralela con memoria compartida. En Proc. Int. Conf. on Parallel and Distributed Information Systems, páginas 226–234.
- Bergsten, B., Couprie, M., y Valduriez, P. (1993). Panorama de arquitecturas paralelas para bases de datos. The Comp. J., 36(8):734–739.
- Berkholz, C., Keppeler, J., y Schweikardt, N. (2017). Respuesta a consultas conjuntivas bajo actualizaciones. En Proc. Simposio ACM SIGACT-SIGMOD sobre Principios de Sistemas de Bases de Datos, páginas 303–318.
- Berlín, J. y Motro, A. (2001). Autoplex: Descubrimiento automatizado de contenido para bases de datos virtuales. En Proc. Int. Conf. sobre Sistemas de Inf. Cooperativa, páginas 108–122.
- Berners-Lee, T. (2006). Datos enlazados. Disponible en <https://www.w3.org/DesignIssues/LinkedData.html>. Último acceso junio de 2019.
- Bernstein, P. y Blaustein, B. (1982). Métodos rápidos para probar aserciones cuantificadas de cálculo relacional. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 39–50.

- Bernstein, P. y Melnik, S. (2007). Gestión de modelos: 2.0: Manipulación de mapeos más completos. En Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos, páginas 1-12.
- Bernstein, P., Blaustein, B. y Clarke, EM (1980a). Mantenimiento rápido de aserciones de integridad semántica mediante datos agregados redundantes. En Actas de la 6.ª Conferencia Internacional sobre Bases de Datos Very, páginas 126-136.
- Bernstein, P., Shipman, P. y Rothnie, JB (1980b). Control de concurrencia en un sistema para bases de datos distribuidas (SDD-1). ACM Trans. Database Syst., 5(1):18-51.
- Bernstein, PA y Chiu, DM (1981). Uso de semiuniones para resolver consultas relacionales. J. ACM, 28 (1):25-40.
- Bernstein, PA y Goodman, N. (1981). Control de concurrencia en sistemas de bases de datos distribuidas. ACM Comput. Surv., 13(2):185-222.
- Bernstein, PA y Goodman, N. (1983). Control de concurrencia multiversión: teoría y algoritmos. ACM Trans. Database Syst., 8(4):465-483.
- Bernstein, PA y Goodman, N. (1984). Un algoritmo para el control y recuperación de concurrencia en bases de datos distribuidas replicadas. ACM Trans. Database Syst., 9(4):596-615.
- Bernstein, PA y Newcomer, E. (1997). Principios del procesamiento de transacciones para los sistemas Profesional. Morgan Kaufmann.
- Bernstein, PA, Goodman, N., Wong, E., Reeve, CL y Jr., JBR (1981). Procesamiento de consultas en un sistema para bases de datos distribuidas (SDD-1). ACM Trans. Database Syst., 6(4):602-625.
- Bernstein, PA, Hadzilacos, V., y Goodman, N. (1987). Control de concurrencia y recuperación en Sistemas de bases de datos. Addison Wesley.
- Bernstein, PA, Giunchiglia, F., Kementsietsidis, A., Mylopoulos, J., Serafini, L. y Zaihrayeu, I. (2002). Gestión de datos para computación peer-to-peer: Una visión. En Actas del 5.º Taller Internacional sobre la World Wide Web y Bases de Datos, páginas 89-94.
- Bernstein, PA, Fekete, A., Guo, H., Ramakrishnan, R. y Tamma, P. (2006). Serialización de concurrencia Relaxed para caché y replicación de nivel medio. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 599-610.
- Beyer, KS, Ercegovac, V., Krishnamurthy, R., Raghavan, S., Rao, J., Reiss, F., Shekita, EJ, Simmen, DE, Tata, S., Vaithyanathan, S. y Zhu, H. (2009). Hacia una plataforma escalable de análisis de contenido empresarial. P. Toro. IEEE TC sobre ingeniería de datos, 32(1):28-35.
- Bharat, K. y Broder, A. (1998). Una técnica para medir el tamaño relativo y la superposición de los motores de búsqueda web públicos. Comp. Networks and ISDN Syst., 30:379-388. (Actas de la 7.ª Conferencia Internacional sobre la World Wide Web).
- Bhowmick, SS, Madria, SK y Ng, WK (2004). Gestión de datos web. Saltador.
- Bifet, A., Gavaldà, R., Holmes, G. y Pfahringer, B. (2018). Aprendizaje automático para flujos de datos: con ejemplos prácticos en MOA. MIT Press.
- Binnig, C., Hildenbrand, S., Färber, F., Kossmann, D., Lee, J. y May, N. (2014). Aislamiento distribuido de instantáneas: las transacciones globales pagan globalmente, las transacciones locales pagan localmente. VLDB J., 23:987-1011.
- Biscondi, N., Brunie, L., Flory, A. y Kosch, H. (1996). Encapsulación del paralelismo intraoperacional en un operador de coincidencia paralela. En Proc. ACPC Conf., volumen 1127 de Lecture Notes in Computer Science, páginas 124-135.
- Bitton, D., Boral, H., DeWitt, DJ, y Wilkinson, W.K. (1983). Algoritmos paralelos para la ejecución de operaciones en bases de datos relacionales. ACM Trans. Database Syst., 8(3):324-353.
- Bitton, D., DeWitt, DJ, Hsiao, D.K. y Menon, J. (1984). Una taxonomía de ordenamiento paralelo. ACM Comput. Surv., 16(3):287-318.
- Bizer, C., Vidal, M.-E., y Skaf-Molli, H. (2018). Datos abiertos enlazados. En Liu, L. y Özsu, MT, editores, Enciclopedia de Sistemas de Bases de Datos, páginas 2096-2101. Springer Nueva York, Nueva York, NY.
- Blanas, S., Patel, J. M., Ercegovac, V., Rao, J., Shekita, E. J. y Tian, Y. (2010). Comparación de algoritmos de unión para el procesamiento de registros en MapReduce. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 975-986.
- Blaustein, B. (1981). Aplicación de aserciones en bases de datos: técnicas y aplicaciones. Tesis doctoral. Universidad de Harvard, Cambridge, Massachusetts.

- Bleiholder, J. y Naumann, F. (2009). Fusión de datos. *ACM Comput. Surv.*, 41(1):1:1–1:41.
- Bonato, A. (2008). Un curso sobre el gráfico web. Sociedad Matemática Americana.
- Bondiombouy, C. y Valduriez, P. (2016). Procesamiento de consultas en sistemas multitienda: una visión general. *Int. J. Computación en la nube*, 5(4):309–346.
- Bondiombouy, C., Kolev, B., Levchenko, O. y Valduriez, P. (2016). Integración de big data multialmacén con CloudMdsQL. *Trans. Sistemas a gran escala centrados en datos y conocimiento*, 28: 48–74.
- Bonifati, A., Summa, G., Pacitti, E. y Draidi, F. (2014). Reformulación de consultas en PDMS basada en la relevancia social. *Trans. Large-Scale Data- and Knowledge-Centered Syst.*, 13:59–90.
- Bonnet, P., Gehrke, J., y Seshadri, P. (2001). Hacia sistemas de bases de datos de sensores. En *Proc. 2.ª edición. Conferencia sobre gestión de datos móviles*, páginas 3–14.
- Bönrström, V., Hinze, A. y Schweppe, H. (2003). Almacenamiento de RDF como grafo. En *Actas del 1.er Congreso Latinoamericano de Web*, páginas 27-36.
- Boral, H. y DeWitt, D. (1983). Máquinas de bases de datos: ¿Una idea que ya pasó de moda? Una crítica del futuro de las máquinas de bases de datos. En *Proc. 3.er Taller Internacional sobre Máquinas de Bases de Datos*, páginas 166–187.
- Boral, H., Alexander, W., Clay, L., Copeland, G., Danforth, S., Franklin, M., Hart, B., Smith, M. y Valduriez, P. (1990). Prototipado de Bubba, un sistema de base de datos altamente paralelo. *IEEE Trans. Conocimiento e Ingeniería de Datos*, 2(1):4–24.
- Borkar, D., Mayuram, R., Sangudi, G. y Carey, MJ (2016). Tenga sus datos y consúltelos también: Del almacenamiento en caché de clave-valor a la gestión de big data. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 239-251.
- Bornea, MA, Dolby, J., Kementsietsidis, A., Srinivas, K., Dantressangle, P., Udrea, O. y Bhattacharjee, B. (2013). Construyendo un almacén RDF eficiente sobre una base de datos relacional. En *Proc. Conferencia internacional ACM SIGMOD sobre gestión de datos*, páginas 121–132.
- Borr, A. (1988). SQL de alto rendimiento mediante integración de sistemas de bajo nivel. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 342–349.
- Bouganin, L., Florescu, D., y Valduriez, P. (1996). Balanceo de carga dinámico en sistemas de bases de datos paralelas jerárquicas. En *Actas de la 22.ª Conferencia Internacional sobre Bases de Datos de Gran Tamaño*, páginas 436–447.
- Bouganin, L., Florescu, D. y Valduriez, P. (1999). Ejecución de consultas multiunión con sesgo en Multiprocesadores NUMA. *Distribuciones paralelas. Bases de datos*, 7(1). En prensa.
- Breibart, Y. y Korth, HF (1997). Replicación y consistencia: Ser perezoso a veces ayuda. En *Proc. ACM SIGACT-SIGMOD Symp. on Principles of Database Systems*, páginas 173-184.
- Breibart, Y. y Silberschatz, A. (1988). Problemas de actualización de múltiples bases de datos. En *Proc. ACM SIGMOD Conferencia Internacional sobre Gestión de Datos*, páginas 135–142.
- Breibart, Y., Olson, PL, y Thompson, GR (1986). Integración de bases de datos en un sistema de bases de datos heterogéneas distribuidas. En *Actas de la 2.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 301–310.
- Brewer, E., Ying, L., Greenfield, L., Cypher, R. y T'so, T. (2016). Discos para centros de datos. Informe técnico, Google.
- Brewer, EA (2000). Hacia sistemas distribuidos robustos (resumen). En *Proc. ACM SIGACT-XIX Simposio SIGOPS sobre los Principios de la Computación Distribuida*, página 7.
- Bright, MW, Hurson, AR y Pakzad, SH (1994). Resolución automatizada de heterogeneidad semántica en múltiples bases de datos. *ACM Trans. Database Syst.*, 19(2):212–253.
- Brill, D., Templeton, M. y Yu, C. (1984). Estrategias de procesamiento distribuido de consultas en MERMAID: Una interfaz para sistemas de gestión de datos. En *Actas de la 1.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 211-218.
- Brin, S. y Page, L. (1998). Anatomía de un motor de búsqueda web hipertextual a gran escala. *Comp. Red*, 30(1-7):107 – 117.
- Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A. y Wiener, J. (2000). Estructura de grafos en la web. *Comp. Netw.*, 33(1-6):309–320.
- Broekstra, J., Kampman, A. y van Harmelen, F. (2002). Sesame: una arquitectura genérica para almacenar y consultar RDF y esquemas RDF. En *Proc. 1º Int. Conferencia sobre web semántica*, páginas 54–68.

- Bu, Y., Howe, B., Balazinska, M. y Ernst, MD (2010). HaLoop: procesamiento iterativo eficiente de datos en grandes clústeres. *Proc. VLDB Endowment*, 3(1):285–296.
- Bu, Y., Howe, B., Balazinska, M. y Ernst, MD (2012). El enfoque HaLoop para la investigación a gran escala. *Análisis iterativo de datos. VLDB J.*, 21(2):169–190.
- Bu, Y., Borkar, VR, Jia, J., Carey, MJ y Condie, T. (2014). Pregelix: Análisis de grafos de mayor tamaño en un motor de flujo de datos. *Proc. VLDB Endowment*, 8(2):161–172.
- Bugiotti, F., Bursztyn, D., Deutsch, A., Ileana, I. y Manolescu, I. (2015). Pegamento invisible: Multitiendas escalables y autoajustables. En *Actas de la 7.ª Conferencia Bienal sobre Investigación Innovadora en Sistemas de Datos*.
- Buneman, P., Davidson, S., Hillebrand, GG y Suciu, D. (1996). Un lenguaje de consulta y técnicas de optimización para datos no estructurados. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 505–516.
- Cahill, MJ, Röhm, U., y Fekete, AD (2009). Aislamiento serializable para bases de datos de instantáneas. *ACM Trans. Database Syst.*, 34(4):Artículo 20.
- Calí, A. y Calvanese, D. (2002). Consulta optimizada de datos integrados en la web. En *Sistemas de información de ingeniería en el contexto de Internet*, páginas 285–301.
- Callan, JP y Connell, ME (2001). Muestreo de bases de datos de texto basado en consultas. *ACM Trans. Sistema de Información*, 19(2):97–130.
- Callan, JP, Connell, ME y Du, A. (1999). Descubrimiento automático de modelos lingüísticos para bases de datos textuales. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 479–490.
- Cammert, M., Krämer, J., Seeger, B. y S. Vaupe. (2006). Un enfoque para la gestión adaptativa de memoria en sistemas de flujo de datos. En *Actas de la 22.ª Conferencia Internacional sobre Ingeniería de Datos*, pág. 137.
- Canaday, RH, Harrison, RD, Ivie, EL, Rydery, JL y Wehr, LA (1974). Un sistema informático de gestión de bases de datos. *Commun. ACM*, 17(10):575–582.
- Cao, P. y Wang, Z. (2004). Problemas de procesamiento de consultas en bases de datos de imágenes (multimedia). En *Proc. ACM SIGACT-SIGOPS 23.º Simposio sobre Principios de Computación Distribuida*, páginas 206-215.
- Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S. y Tzoumas, K. (2015). Apache Flink™: procesamiento por lotes y en streaming en un solo motor. *P. Toro. IEEE TC sobre ingeniería de datos*, 38 (4): 28–38.
- Carey, M. y Lu, H. (1986). Balanceo de carga en un sistema de base de datos distribuido localmente. En *Proc. Conferencia internacional ACM SIGMOD sobre gestión de datos*, páginas 108–119.
- Castano, S. y Antonellis, VD (1999). Un entorno de herramientas de análisis y conciliación de esquemas para bases de datos heterogéneas. En *Actas de la 3.ª Conferencia Internacional sobre Ingeniería de Bases de Datos y Aplicaciones*, páginas 53-62.
- Castaño, S., Fugini, MG, Martella, G. y Samarati, P. (1995). *Seguridad de la base de datos*. addison Wesley.
- Castro, M. y Liskov, B. (1999). Tolerancia práctica a fallas bizantinas. En *Actas del 3.er Simposio USENIX sobre diseño e implementación de sistemas operativos*, páginas 173–186.
- Cellary, W., Gelenbe, E., y Morzy, T. (1988). Control de concurrencia en bases de datos distribuidas. *Sistemas. Holanda Septentrional*.
- Ceri, S. y Owicki, S. (1982). Sobre el uso de métodos optimistas para el control de concurrencia en bases de datos distribuidas. En *Proc. 6.º Taller de Berkeley sobre Gestión de Datos Distribuidos y Redes de Computadoras*, págs. 117-130.
- Ceri, S. y Pelagatti, G. (1983). Corrección de las estrategias de ejecución de consultas en bases de datos distribuidas. *ACM Trans. Base de datos Syst.*, 8(4):577–607.
- Ceri, S. y Pernici, B. (1985). DATAID–D: Metodología para el diseño de bases de datos distribuidas. En Albano, V. d. A. y di Leva, A., editores, *Diseño de Bases de Datos Asistido por Computadora*, páginas 157–183. Holanda del Norte.
- Ceri, S. y Widom, J. (1993). Gestión de la heterogeneidad semántica con reglas de producción y colas persistentes. En *Actas de la 19.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 108-119.
- Ceri, S., Martella, G. y Pelagatti, G. (1982a). Asignación óptima de archivos en una red informática: A Método de solución basado en el problema de la mochila. *Comp. Netw.*, 6:345–357.
- Ceri, S., Negri, M. y Pelagatti, G. (1982b). Partición horizontal de datos en el diseño de bases de datos. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*, páginas 128–136.



- Ceri, S., Navathe, SB, y Wiederhold, G. (1983). Diseño de distribución de bases de datos lógicas. esquemas. Traducción IEEE. Software. Ing., SE-9(4):487–503.
- Ceri, S., Gottlob, G. y Pelagatti, G. (1986). Taxonomía y propiedades formales de uniones distribuidas. Inf. Syst., 11(1):25–40.
- Ceri, S., Pernici, B. y Wiederhold, G. (1987). Metodologías de diseño de bases de datos distribuidas. Proc. IEEE, 75(5):533–546.
- Chairunnanda, P., Daudjee, K. y Özsu, MT (2014). ConfluxDB: replicación multimaestro para bases de datos particionadas con aislamiento de instantáneas. Proc. VLDB Endowment, 7(11):947–958.
- Chakrabarti, K., Keogh, E., Mehrotra, S. y Pazzani, M. (2002). Reducción de dimensionalidad localmente adaptativa para la indexación de bases de datos de series temporales extensas. ACM Trans. Database Syst., 27.
- Chakrabarti, S., Dom, B. e Indyk, P. (1998). Clasificación mejorada de hipertextos mediante hipervínculos. En Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos, páginas 307 – 318.
- Chamberlin, D. (2018). SQL++ para usuarios de SQL: Un tutorial. CouchBase Inc.
- Chamberlin, D., Gray, J. y Traiger, I. (1975). Vistas, autorización y bloqueo en un sistema de bases de datos relacionales. En Proc. National Computer Conf, páginas 425–430.
- Chambers, C., Raniwala, A., Perry, F., Adams, S., Henry, RR, Bradshaw, R. y Weizenbaum, N. (2010). FlumeJava: pipelines de datos paralelos fáciles y eficientes. En Proc. Conferencia ACM SIGPLAN 2010 sobre Diseño e Implementación de Lenguajes de Programación, páginas 363–375.
- Chandra, TD, Griesemer, R. y Redstone, J. (2007). Paxos en vivo: Una perspectiva de ingeniería. En Proc. 26.º Simposio ACM SIGACT-SIGOPS sobre los Principios de la Computación Distribuida, páginas 398–407.
- Chandrasekaran, S., Cooper, O., Deshpande, A., Franklin, MJ, Hellerstein, J.M., Hong, W., Krishnamurthy, S., Madden, S., Raman, V., Reiss, F. y Shah, MA (2003). TelegraphCQ: Procesamiento continuo de flujo de datos para un mundo incierto. En Actas de la 1.ª Conferencia Bienal sobre Investigación Innovadora en Sistemas de Datos.
- Chang, F., Dean, J., Ghemawat, S., Hsieh, W., Wallach, D., Burrows, M., Chandra, T., Fikes, A. y Gruber, R. (2008). Bigtable: Un sistema de almacenamiento distribuido para datos estructurados. ACM Trans. Comp. Syst., 26(2):Artículo 4.
- Chang, SK y Liu, AC (1982). Asignación de archivos en una base de datos distribuida. Int. J. Comput. Inf. Ciencia, 11(5):325–340.
- Chattopadhyay, B., Lin, L., Liu, W., Mittal, S., Aragonda, P., Lychagina, V., Kwon, Y. y Wong, M. (2011). Tenzing: Una implementación de SQL en el marco MapReduce. Proc. VLDB Endowment, 4(12):1318–1327.
- Chaudhuri, S., Ganjam, K., Ganti, V. y Motwani, R. (2003). Emparejamiento difuso robusto y eficiente para la limpieza de datos en línea. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 313–324.
- Chen, R., Shi, J., Chen, Y. y Chen, H. (2015). PowerLyra: Cálculo de grafos diferenciados y partición en grafos sesgados. En Actas de la 10.ª Conferencia Europea ACM SIGOPS/EuroSys sobre Sistemas Computacionales, páginas 1:1–1:15.
- Chiu, DM y Ho, YC (1980). Una metodología para interpretar consultas de árbol en expresiones de semiunión óptimas. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 169–178.
- Cho, J. y García-Molina, H. (2000). La evolución de la web y sus implicaciones para un rastreador incremental. En Actas de la 26.ª Conferencia Internacional sobre Bases de Datos de Gran Tamaño.
- Cho, J. y Ntoulas, A. (2002). Detección efectiva de cambios mediante muestreo. En Actas de la 28.ª Conferencia Internacional sobre Bases de Datos Muy Grandes.
- Cho, J., Garcia-Molina, H., y Page, L. (1998). Rastreo eficiente mediante ordenación de URL. Comp. Redes y sistemas RDSI, 30(1-7):161–172.
- Chockler, G., Keidar, I. y Vitenberg, R. (2001). Especificaciones de comunicación grupal: un estudio exhaustivo. ACM Comput. Surv., 33(4):427–469.
- Chong, E., Das, S., Eadon, G. y Srinivasan, J. (2005). Un esquema eficiente de consultas RDF basado en SQL. En Actas de la 31.ª Conferencia Internacional sobre Bases de Datos Muy Grandes, páginas 1216–1227.
- Chu, WW (1969). Asignación óptima de archivos en un sistema multicomputador. IEEE Trans. Comput., C-18(10):885–889.

- Chu, WW (1973). Asignación óptima de archivos en una red informática. En Abramson, N. y Kuo, FF, editores, *Redes de comunicación informática*, páginas 82–94.
- Chu, WW (1976). Rendimiento de los sistemas de directorio de archivos para bases de datos en estrella y distribuidas. *redes*. En *Proc. National Computer Conf.*, volumen 45, páginas 577–587.
- Chu, WW y Nahouraii, EE (1975). Consideraciones sobre el diseño de directorios de archivos para bases de datos distribuidas. En *Actas de la 1.ª Conferencia Internacional sobre Bases de Datos Very*, páginas 543–545.
- Chu, X., Ilyas, IF, y Papotti, P. (2013). Descubrimiento de restricciones de negación. *Proc. VLDB Dotación*, 6(13):1498–1509.
- Chundi, P., Rosenkrantz, DJ, y Ravi, SS (1996). Actualizaciones diferidas y ubicación de datos en bases de datos distribuidas. En *Proc. ACM SIGACT-SIGMOD Symp. on Principles of Database Systems*, páginas 469–476.
- Civelek, FN, Dogac, A. y Spaccapietra, S. (1988). Un enfoque de sistemas expertos para la definición e integración de vistas. En *Actas de la 7.ª Conferencia Internacional sobre el Enfoque Entidad-Relación*, páginas 229–249.
- Cohen, J. (2009). Manipulación de gráficos en un mundo MapReduce. *Computación en Ciencias e Ingeniería*. 11(4):29–41.
- Cole, RL y Graefe, G. (1994). Optimización de planes de evaluación de consultas dinámicas. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 150–160.
- Coletta, R., Castanier, E., Valdúriez, P., Frisch, C., Ngo, D. y Bellahsene, Z. (2012). Integración de datos públicos con Websmatch. En *Proc. Int. Workshop on Open Data*, páginas 5–12.
- Copeland, G., Alexander, W., Boughter, E. y Keller, T. (1988). Ubicación de datos en bubba. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*, páginas 99–108.
- Corbett, JC, Dean, J., Epstein, M., Fikes, A., Frost, C., Furman, JJ, Ghemawat, S., Gubarev, A., Heiser, C., Hochschild, P., Hsieh, W., Kanthak, S., Kogan, E., Li, H., Lloyd, A., Melnik, S., Mwaura, D., Nagle, D., Quinlan, S., Rao, R., Rolig, L., Saito, Y., Szymaniak, M., Taylor, C., Wang, R. y Woodford, D. (2013). Spanner: la base de datos distribuida globalmente de Google. *Transmisión ACM. Sistema de base de datos*, 31(3):8:1–8:22.
- Crainiceanu, A., Linga, P., Gehrke, J. y Shanmugasundaram, J. (2004). Consultas en redes peer-to-peer mediante p-árboles. En *Actas del 7.º Taller Internacional sobre la World Wide Web y Bases de Datos*, páginas 25–30.
- Cranor, C., Johnson, T., Spatscheck, O. y Shkapenyuk, V. (2003). Gigascope: Monitoreo de redes de alto rendimiento con una interfaz SQL. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 647–651.
- Crespo, A. y García-Molina, H. (2002). Índices de enrutamiento para sistemas peer-to-peer. En *Proc. 22.ª edición. IEEE Int. Conf. sobre sistemas de computación distribuida*, páginas 23–33.
- Cuenca-Acuña, F., Peery, C., Martin, R. y Nguyen, T. (2003). PlanetP: uso de la comunicación para construir comunidades de intercambio de información entre pares con acceso a contenido. En *el Simposio Internacional IEEE sobre Computación Distribuida de Alto Rendimiento*, páginas 236–249.
- Curino, C., Jones, E., Zhang, Y. y Madden, S. (2010). Schism: un enfoque basado en la carga de trabajo para Replicación y particionamiento de bases de datos. *Proc. VLDB Endowment*, 3(1):48–57.
- Curino, C., Jones, EPC, Madden, S. y Balakrishnan, H. (2011). Monitoreo y consolidación de bases de datos con capacidad de carga de trabajo. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 313–324.
- Cusumano, MA (2010). Computación en la nube y SaaS como nuevas plataformas informáticas. *Commun. ACM*, 53(4):27–29.
- Dasgupta, S., Coakley, K. y Gupta, A. (2016). Ingesta y derivación de datos basada en analítica en el polystore AWESOME. En *Proc. 2016 IEEE Int. Conf. on Big Data*, páginas 2555–2564.
- Daswani, N., García-Molina, H., y Yang, B. (2003). Problemas abiertos en el intercambio de datos entre pares. *sistemas*. En *Actas de la 9.ª Conferencia Internacional sobre Teoría de Bases de Datos*, páginas 1–15.
- Daudjee, K. y Salem, K. (2004). Replicación perezosa de bases de datos con garantías de orden. En *Proc. 20ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 424–435.
- Daudjee, K. y Salem, K. (2006). Replicación diferida de bases de datos con aislamiento de instantáneas. En *Actas de la 32.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 715–726.
- Davenport, RA (1981). Diseño de sistemas de bases de datos distribuidas. *Comp. J.*, 24(1):31–41.

- Davidson, SB (1984). Optimismo y consistencia en sistemas de bases de datos distribuidas y particionadas. *ACM Trans. Base de datos Syst.*, 9(3):456–481.
- Davidson, SB, Garcia-Molina, H., y Skeen, D. (1985). Consistencia en redes particionadas. *ACM Comput. Surv.*, 17(3):341–370.
- Dawson, JL (1980). Un modelo de demanda de usuarios para el diseño de bases de datos distribuidas. En *Compendio de Artículos – COMPCON*, páginas 211–216.
- Dayal, U. y Bernstein, P. (1978). Sobre la actualizabilidad de las vistas relacionales. En *Actas de la 4.ª Conferencia Internacional en Very Data Bases*, páginas 368–377.
- Dayal, U. y Hwang, H. (1984). Definición de vista y generalización para la integración de bases de datos en MULTIBASE: Un sistema para bases de datos distribuidas heterogéneas. *IEEE Trans. Softw. Eng.*, SE-10(6):628–644.
- Dean, J. y Ghemawat, S. (2004). MapReduce: Procesamiento de datos simplificado en clústeres grandes. En *Actas del 6.º Simposio USENIX sobre Diseño e Implementación de Sistemas Operativos*, páginas 137–149.
- Dean, J. y Ghemawat, S. (2010). MapReduce: una herramienta flexible de procesamiento de datos. *Commun. ACM*, 53(1):72–77.
- DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubra-manian, S., Vosshall, P. y Vogels, W. (2007). Dynamo: El almacén de clave-valor de alta disponibilidad de Amazon. En *Proc. XXI Simposio ACM sobre Principios de Sistemas Operativos*, páginas 205-220.
- Demers, AJ, Greene, DH, Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, HE, Swinehart, DC y Terry, DB (1987). Algoritmos epidémicos para el mantenimiento de bases de datos replicadas. En *Proc. ACM SIGACT-SIGOPS, 6.º Simposio sobre los Principios de la Computación Distribuida*, páginas 1-12.
- Deshpande, A. y Gupta, A. (2018). Principios de la gestión y el análisis de datos de grafos. *ACM Books*. Próximamente.
- Devine, R. (1993). Diseño e implementación de DDH: Un algoritmo hash dinámico distribuido. En *Proc. 4ta Conferencia Internacional sobre Fundamentos de Organización de Datos y Algoritmos*, páginas 101–114.
- Dewitt, D. y Stonebraker, M. (2009). MapReduce: Un gran paso atrás. [https://homes.cs.washington.edu/~billhowe/mapreduce\\_a\\_major\\_step\\_backwards.html](https://homes.cs.washington.edu/~billhowe/mapreduce_a_major_step_backwards.html).
- DeWitt, D., Naughton, J., Schneider, D. y Seshadri, S. (1992). Manejo práctico de sesgos en Uniones paralelas. En *Actas de la 22.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 27–40.
- DeWitt, DJ y Gerber, R. (1985). Algoritmos de unión multiprocesador basados en hash. En *Proc. 11.ª edición. Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 151–164.
- DeWitt, DJ y Gray, J. (1992). Sistemas de bases de datos paralelas: El futuro del alto rendimiento. *sistemas de bases de datos. Commun. ACM*, 35(6):85–98.
- DeWitt, DJ, Katz, R., Olken, F., Shapiro, L., Stonebraker, M. y Wood, D. (1984). Técnicas de implementación para sistemas de bases de datos en memoria principal. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 1–8.
- DeWitt, DJ, Gerber, RH, Graek, G., Heytens, ML, Kumar, KB y Muralikrishna, M. (1986). Gamma: Una máquina de base de datos de flujo de datos de alto rendimiento. En *Actas de la 12.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 228-237.
- DeWitt, DJ, Paulson, E., Robinson, E., Naughton, J., Royalty, J., Shankar, S. y Krioukov, A. (2008). Clustera: un sistema integrado de computación y gestión de datos. *Proc. Fundación VLDB*, 1:28–41.
- DeWitt, DJ, Halverson, A., Nehme, RV, Shankar, S., Aguilar-Saborit, J., Avanes, A., Flasz, M. y Gramling, J. (2013). Procesamiento de consultas divididas en Polybase. En *Proc. ACM SIGMOD Int. Conferencia sobre gestión de datos*, páginas 1255–1266.
- Dhamankar, R., Lee, Y., Doan, A., Halevy, AY, y Domingos, P. (2004). iMAP: Descubrimiento de mapeos complejos entre esquemas de bases de datos. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 383–394.
- Ding, L. y Rundensteiner, E. (2004). Evaluación de uniones de ventanas sobre flujos puntuados. En *Actas de la 13.ª Conferencia Internacional de la ACM sobre Gestión de la Información y el Conocimiento*, páginas 98-107.
- Ding, L., Mehta, N., Rundensteiner, E. y Heineman, G. (2004). Unión de flujos puntuados. En *Avances en tecnología de bases de datos, Actas de la 9.ª Conferencia Internacional sobre la extensión de la tecnología de bases de datos*, páginas 587–604.

- Dinh, TTA, Liu, R., Zhang, M., Chen, G., Ooi, BC y Wang, J. (2018). Desenredando la cadena de bloques: Una perspectiva de procesamiento de datos de los sistemas de cadena de bloques. *IEEE Trans. Knowl. and Data Eng.*, 30(7):1366–1385.
- Do, H. y Rahm, E. (2002). COMA: un sistema para la combinación flexible de enfoques de coincidencia de esquemas. En *Actas de la 28.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 610–621.
- Doan, A. y Halevy, AY (2005). Investigación de integración semántica en la comunidad de bases de datos: Un breve estudio. *AI Magazine*, 26(1):83–94.
- Doan, A., Domingos, P. y Halevy, AY (2001). Conciliación de esquemas de fuentes de datos dispares: Un enfoque de aprendizaje automático. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 509–520.
- Doan, A., Domingos, P. y Halevy, A. (2003a). Aprendizaje para la correspondencia de los esquemas de las fuentes de datos: Un enfoque multiestrategia. *Aprendizaje Automático*, 50(3):279–301.
- Doan, A., Madhavan, J., Dhamankar, R., Domingos, P. y Halevy, A. (2003b). Aprendizaje para la correspondencia de ontologías en la web semántica. *VLDB J.*, 12(4):303–319.
- Doan, A., Halevy, A., e Ives, Z. (2012). Principios de integración de datos. Morgan Kaufmann.
- Dogac, A., Kalinichenko, L., Özsu, MT, y Sheth, A., editores. (1998). *Avances en sistemas de flujo de trabajo e interoperabilidad*. Springer.
- Dong, XL y Naumann, F. (2009). Fusión de datos: resolución de conflictos de datos para la integración. *Proc. Dotación VLDB*, 2(2):1654–1655.
- Dong, XL y Srivastava, D. (2015). Integración de Big Data. *Conferencias de síntesis sobre gestión de datos*. Morgan & Claypool Publishers.
- Dong, XL, Berti-Equille, L., y Srivastava, D. (2009a). Descubrimiento de la verdad y detección de copias en un mundo dinámico. *Proc. VLDB Endowment*, 2(1):562–573.
- Dong, XL, Berti-Equille, L., y Srivastava, D. (2009b). Integración de datos conflictivos: el rol de la dependencia de la fuente. *Proc. VLDB Endowment*, 2(1):550–561.
- Dowdy, LW y Foster, DV (1982). Modelos comparativos del problema de asignación de archivos. *ACM Comput. Surv.*, 14(2):287–313.
- Du, W., Krishnamurthy, R. y Shan, M. (1992). Optimización de consultas en un SGBD heterogéneo. En *Proc. 18a Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 277–291.
- Du, W., Shan, M. y Dayal, U. (1995). Reducción del tiempo de respuesta de consultas multibase de datos mediante el balanceo de árboles. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 293–303.
- Duggan, J., Elmore, AJ, Stonebraker, M., Balazinska, M., Howe, B., Kepner, J., Madden, S., Maier, D., Mattson, T. y Zdonik, SB (2015). El sistema de almacenamiento en múltiples capas BigDAWG. *ACM SIGMOD Rec.*, 44(2):11–16.
- Duschka, OM y Genesereth, MR (1997). Respuesta a consultas recursivas mediante vistas. En *Proc. Simposio ACM SIGACT-SIGMOD sobre principios de sistemas de bases de datos*, páginas 109–116.
- Eager, DL y Sevcik, KC (1983). Logrando robustez en sistemas de bases de datos distribuidas. *ACM Trans. Sistema de bases de datos*, 8(3):354–381.
- Edwards, J., McCurley, K. y Tomlin, J. (2001). Un modelo adaptativo para optimizar el rendimiento. de un rastreador web incremental. En *Actas de la 10.ª Conferencia Internacional sobre la World Wide Web*.
- El Abbadi, A., Skeen, D. y Cristian, F. (1985). Un protocolo eficiente y tolerante a fallos para la gestión de datos replicados. En *Proc. ACM SIGACT-SIGMOD Symp. on Principles of Database Systems*, páginas 215–229.
- Elbushra, MM y Lindström, J. (2015). Bases de datos causales consistentes. *Revista Abierta de Bases de datos*, 2(1):17–35.
- Elmagarmid, A., Rusinkiewicz, M., y Sheth, A., editores. (1999). *Gestión de heterogeneidad y sistemas de bases de datos autónomos*. Morgan Kaufmann.
- Elmagarmid, AK (1986). Un estudio de algoritmos distribuidos de detección de interbloques. *ACM SIGMOD Rec.*, 15(3):37–45.
- Elmagarmid, AK, editor. (1992). *Modelos de transacción para aplicaciones avanzadas de bases de datos*. Morgan Kaufmann.
- Elmagarmid, AK, Soundararajan, N., y Liu, MT (1988). Un algoritmo distribuido de detección y resolución de interbloques y su prueba de corrección. *IEEE Trans. Softw. Eng.*, 14(10):1443–1452.

- Elmasri, R., Larson, J. y Navathe, SB (1987). Algoritmos de integración para bases de datos y diseño lógico de bases de datos. Informe técnico, Centro de Investigación Corporativa de Honeywell, Golden Valley, Minnesota.
- Elmore, AJ, Arora, V., Taft, R., Pavlo, A., Agrawal, D. y El Abbadi, A. (2015). Squall: Reconfiguración en vivo de grano fino para bases de datos de memoria principal particionadas. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 299–313.
- Elseidy, M., Elguindy, A., Vitorovic, A. y Koch, C. (2014). Uniones en línea escalables y adaptativas. Proc. Dotación VLDB, 7(6):441–452.
- Embley, DW, Jackman, D., y Xu, L. (2001). Explotación multifacética de metadatos para el descubrimiento de coincidencias de atributos en la integración de información. En Proc. Taller sobre Integración de Información en la Web, páginas 110–117.
- Embley, DW, Jackman, D., y Xu, L. (2002). Descubrimiento de coincidencias de atributos en la integración de información: aprovechamiento de múltiples facetas de los metadatos. Revista de la Sociedad Brasileña de Computación, 8(2):32–43.
- Epstein, R., Stonebraker, M. y Wong, E. (1978). Procesamiento de consultas en un sistema de bases de datos relacionales distribuidas. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 169–180.
- Eswaran, KP (1974). Ubicación de registros en un archivo y asignación de archivos en una red informática. En Procesamiento de la Información '74, págs. 304–307.
- Etzion, O. y Niblett, P. (2010). Procesamiento de eventos en acción. Manning.
- Evrendilek, C., Dogac, A., Nural, S. y Ozcan, F. (1997). Optimización de consultas multibase de datos. Distribuidor Paralelo de Bases de Datos, 5(1):77–114.
- Eyal, I., Gencer, AE, Sirer, EG, y van Renesse, R. (2016). Bitcoin-ng: Un protocolo de cadena de bloques escalable. En Actas del 13.º Simposio USENIX sobre Diseño e Implementación de Sistemas en Red, páginas 45–59.
- Fagin, R. (2002). Combinación de información difusa: una visión general. ACM SIGMOD Rec., 31(2):109–118.
- Fagin, R., Lotem, J. y Naor, M. (2003). Algoritmos de agregación óptimos para middleware. Revista de Ciencias de la Computación y Sistemas, 66(4):614–656.
- Fagin, R., Kolaitis, PG, Miller, RJ y Popa, L. (2005). Intercambio de datos: semántica y respuesta a consultas. Theor. Comp. Sci., 336(1):89–124.
- Faleiro, JM y Abadi, DJ (2015). Replantando el control de concurrencia multiversión serializable. Proc. Dotación VLDB, 8(11):1190–1201.
- Faloutsos, C. y Christodoulakis, S. (1984). Archivos de firmas: un método de acceso a documentos y su evaluación analítica. ACM Trans. Information Syst., 2(4):267–288.
- Farid, MH, Roatis, A., Ilyas, IF, Hoffmann, H. y Chu, X. (2016). CLAMS: aportando calidad a los lagos de datos. En Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos, páginas 2089–2092.
- Farrag, AA y Özsu, MT (1989). Utilizando el conocimiento semántico de las transacciones para aumentar concurrencia. ACM Trans. Database Syst., 14(4):503–525.
- Fekete, A., Lynch, N., Merritt, M. y Weihl, W. (1987a). Transacciones anidadas y bloqueo de lectura/escritura. Memorandum técnico MIT/LCS/TM–324, Instituto Tecnológico de Massachusetts, Cambridge, Massachusetts.
- Fekete, A., Lynch, N., Merritt, M. y Weihl, W. (1987b). Transacciones anidadas, bloqueo basado en conflictos y atomicidad dinámica. Memorandum técnico MIT/LCS/TM–340, Instituto Tecnológico de Massachusetts, Cambridge, Massachusetts.
- Fekete, A., Lynch, N., Merritt, M. y Weihl, W. (1989). Bloqueo basado en conmutatividad para transacciones anidadas. Memorandum técnico MIT/LCS/TM–370b, Instituto Tecnológico de Massachusetts, Cambridge, Massachusetts.
- Fernández, M., Florescu, D., y Levy, A. (1997). Un lenguaje de consulta para un sistema de gestión de sitios web. ACM SIGMOD Rec., 26(3):4–11.
- Fernandez, RC, Migliavacca, M., Kalyvianaki, E. y Pietzuch, P. (2013). Integración del escalamiento horizontal y la tolerancia a fallos en el procesamiento de flujos mediante la gestión del estado del operador. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 725–736.

- Fernández-Moctezuma, R., Tufte, K., y Li, J. (2009). Retroalimentación interoperador en sistemas de gestión de flujos de datos mediante puntuación. En *Actas de la 4.ª Conferencia Bial sobre Investigación Innovadora en Sistemas de Datos*.
- Ferraiolo, D. y Kuhn, R. (1992). Control de acceso basado en roles. En *Proc. National Computer Conf.*, páginas 554–563.
- Fisher, MK y Hochbaum, DS (1980). Ubicación de bases de datos en redes informáticas. *J. ACM*, 27 (4):718–735.
- Fisher, PS, Hollist, P., y Slonim, J. (1980). Una metodología de diseño para bases de datos distribuidas. En *Digest of Papers – COMPCON*, páginas 199–202.
- Florentin, JJ (1974). Auditoría de consistencia de bases de datos. *Comp. J.*, 17(1):52–58.
- Florescu, D., Levy, A. y Mendelzon, A. (1998). Técnicas de bases de datos para la World Wide Web: un estudio. *ACM SIGMOD Rec.*, 27(3):59–74.
- Friedman, M., Levy, AY, y Millstein, TD (1999). Planes de navegación para la integración de datos. En *Actas de la 16.ª Conferencia Nacional sobre Inteligencia Artificial y de la 11.ª Conferencia sobre Aplicaciones Innovadoras de la Inteligencia Artificial*, páginas 67–73.
- Fu, Y., Ong, KW, Papakonstantinou, Y., y Zamora, E. (2014). FORWARD: Interfaces de usuario centradas en datos que utilizan plantillas declarativas que integran eficientemente componentes JavaScript de terceros. *Proc. Dotación VLDB*, 7(13):1649–1652.
- Furtado, C., Lima, AAB, Pacitti, E., Valduriez, P. y Mattoso, M. (2008). Particionamiento híbrido adaptativo para el procesamiento de consultas OLAP en un clúster de bases de datos. *Revista Internacional de Computación y Redes de Alto Rendimiento*, 5(4):251–262.
- Fushimi, S., Kitsuregawa, M. y Tanaka, H. (1986). Resumen del software de sistema de una máquina de base de datos relacional paralela GRACE. En *Actas de la 12.ª Conferencia Internacional sobre Bases de Datos de Gran Tamaño*, páginas 209–219.
- Gadepally, V., Chen, P., Duggan, J., Elmore, A. J., Haynes, B., Kepner, J., Madden, S., Mattson, T. y Stonebraker, M. (2016). El sistema y la arquitectura de almacenamiento múltiple BigDAWG. En *Proc. IEEE High Performance Extreme Computing Conf.*, páginas 1–6.
- Galhardas, H., Florescu, D., Shasha, D., Simon, E. y Saita, C.-A. (2001). Limpieza declarativa de datos: Lenguaje, modelo y algoritmos. En *Actas de la 27.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 371–380.
- Gançarski, S., Naacke, H., Pacitti, E. y Valduriez, P. (2007). El sistema leganet: Frescura Enrutamiento de transacciones consciente en un clúster de bases de datos. *Inf. Syst.*, 32(7):320–343.
- Ganesan, P., Yang, B. y García-Molina, H. (2004). Un toro para gobernarlos a todos: Consultas multidimensionales en sistemas P2P. En *Actas del 7.º Taller Internacional sobre la World Wide Web y Bases de Datos*, páginas 19–24.
- Gankidi, VR, Teletia, N., Patel, J. M., Halverson, A. y DeWitt, DJ (2014). Indexación de datos HDFS en PDW: separación de los datos del índice. *Proc. VLDB Endowment*, 7(13):1520–1528.
- García-Molina, H. (1982). Elecciones en sistemas de computación distribuida. *IEEE Trans. Comput.*, C-31(1):48–59.
- García-Molina, H. (1983). Uso del conocimiento semántico para el procesamiento de transacciones en un sistema distribuido. base de datos. *ACM Trans. Database Syst.*, 8(2):186–213.
- García-Molina, H. y Salem, K. (1987). Sagas. En *Actas de la Conferencia Internacional sobre Gestión de la ACM SIGMOD. de Datos*, páginas 249–259.
- García-Molina, H. y Wiederhold, G. (1982). Transacciones de sólo lectura en una base de datos distribuida. *ACM Trans. Base de datos Syst.*, 7(2):209–234.
- García-Molina, H., Gawlick, D., Klein, J., Kleissner, K. y Salem, K. (1990). Coordinación de actividades multitransacciones. Informe Técnico CS-TR-247-90, Departamento de Ciencias de la Computación, Universidad de Princeton.
- García-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J. D., Vassalos, V. y Widom, J. (1997). El enfoque TSIMMIS para la mediación: Modelos y lenguajes de datos. *J. Intell. Information Syst.*, 8(2):117–132.
- Garofalakis, MN y Ioannidis, YE (1996). Programación multidimensional de recursos para consultas paralelas. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 365–376.

- Gavish, B. y Pirkul, H. (1986). Ubicación de computadoras y bases de datos en sistemas informáticos distribuidos. *IEEE Trans. Comput.*, C-35(7):583–590.
- Gedik, B. (2014). Funciones de partición para el paralelismo de datos con estado en el procesamiento de flujos. *VLDB J.*, 23:517–539.
- Georgakopoulos, D., Hornick, M. y Sheth, A. (1995). Una visión general de la gestión del flujo de trabajo: Del modelado de procesos a la infraestructura de automatización del flujo de trabajo. *Distributed Parall. Databases*, 3: 119–153.
- Ghemawat, S., Gobioff, H. y Leung, S. (2003). El sistema de archivos de Google. En *Actas del 19.º ACM. Sim. sobre principios de sistemas operativos*, páginas 29–43.
- Ghoting, A., Krishnamurthy, R., Pednault, EPD, Reinwald, B., Sindhvani, V., Tatikonda, S., Tian, Y. y Vaithyanathan, S. (2011). SystemML: Aprendizaje automático declarativo en MapReduce. En *Actas de la 27.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 231–242.
- Gifford, DK (1979). Votación ponderada para datos replicados. En *Actas del 7.º Simposio ACM sobre Operaciones. Principios del sistema*, páginas 50–159.
- Gilbert, S. y Lynch, NA (2002). La conjetura de Brewer y la viabilidad de servicios web consistentes, disponibles y con tolerancia a particiones. *SIGACT News*, 33(2):51–59.
- Glasbergen, B., Abebe, M., Daudjee, K., Foggo, S. y Pacaci. (2018). Apollo: Aprendizaje de correlaciones de consultas para el almacenamiento en caché predictivo en sistemas geodistribuidos. En *Actas de la 21.ª Conferencia Internacional sobre la Extensión de la Tecnología de Bases de Datos*, páginas 253-264.
- Golab, L. y Özsu, MT (2003). Procesamiento de uniones múltiples de ventana deslizante en consultas continuas sobre flujos de datos. En *Actas de la 29.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 500–511.
- Golab, L. y Özsu, MT (2010). Sistemas de flujo de datos. Síntesis de conferencias sobre gestión de datos. Morgan y Claypool.
- Goldman, KJ (1987). Replicación de datos en sistemas transaccionales anidados. Informe técnico MIT/LCS/TR-390, Instituto Tecnológico de Massachusetts, Cambridge, Massachusetts.
- Goldman, R. y Widom, J. (1997). Guías de datos: Facilitación de la formulación y optimización de consultas en bases de datos semiestructuradas. En *Actas de la 23.ª Conferencia Internacional sobre Bases de Datos de Gran Tamaño*, páginas 436–445.
- Gonzalez, JE, Low, Y., Gu, H., Bickson, D. y Guestrin, C. (2012). PowerGraph: Computación distribuida grafo-paralela en grafos naturales. En *Proc. 10.º Simposio USENIX sobre Diseño e Implementación de Sistemas Operativos*, páginas 17-30.
- Gonzalez, JE, Xin, RS, Dave, A., Crankshaw, D., Franklin, MJ y Stoica, I. (2014). GraphX: procesamiento de grafos en un marco de flujo de datos distribuido. En *Actas del 11.º Simposio USENIX sobre Diseño e Implementación de Sistemas Operativos*, páginas 599–613.
- Goodman, JR y Woest, PJ (1988). El multicubo de Wisconsin: Un nuevo multiprocesador coherente con caché a gran escala. Informe técnico TR766, Universidad de Wisconsin-Madison.
- Gounaris, A., Paton, NW, Fernandes, AAA y Sakellariou, R. (2002). Procesamiento adaptativo de consultas: Un estudio. En *Proc. Conferencia Nacional Británica sobre Bases de Datos*, págs. 11-25.
- Graefe, G. (1990). Encapsulación del paralelismo en los sistemas de procesamiento de consultas Volcano. En *Proc. Conferencia internacional ACM SIGMOD sobre gestión de datos*, páginas 102–111.
- Graefe, G. (1993). Técnicas de evaluación de consultas para grandes bases de datos. *ACM Comput. Surv.*, 25(2): 73–170.
- Graefe, G. (1994). Volcano: un sistema de evaluación de consultas extensible y paralelo. *IEEE Trans. Conocimiento e Ingeniería de Datos*, 6(1):120–135.
- Graefe, G. y Kuno, H. (2010a). Índices autoseleccionables, autoajustables y optimizados incrementalmente. En *Actas de la 13.ª Conferencia Internacional sobre la ampliación de la tecnología de bases de datos*, páginas 371–381.
- Graefe, G. y Kuno, H. (2010b). Indexación adaptativa para claves relacionales. En *Proc. Talleres de 26ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 69–74.
- Graefe, G., Idreos, S., Kuno, H. y Manegold, S. (2010). Benchmarking de indexación adaptativa. En *Proc. Conferencia Tecnológica TPC sobre Evaluación del Rendimiento, Medición y Caracterización de Sistemas Complejos*, páginas 169–184.
- Graefe, G., Halim, F., Idreos, S., Kuno, H. y Manegold, S. (2012). Control de concurrencia para indexación adaptativa. *Proc. VLDB Endowment*, 5(7):656–667.

- Graefe, G., Halim, F., Idreos, S., Kuno, HA, Manegold, S. y Seeger, B. (2014). Transaccional Soporte para indexación adaptativa. *Vldb J.*, 23(2):303–328.
- Grapa, E. y Belford, GG (1977). Algunos teoremas para facilitar la resolución del problema de asignación de archivos. *Comunitario. ACM*, 20(11):878–882.
- Gravano, L., Garcia-Molina, H., y Tomasic, A. (1999). Glosa: Descubrimiento de fuentes textuales a lo largo de la Internet. *ACM Trans. Database Syst.*, 24(2):229–264.
- Gray, J. (1979). Notas sobre sistemas operativos de bases de datos. En Bayer, R., Graham, R., y Seegmüller, G., editores, *Sistemas operativos: un curso avanzado*, páginas 393–481. Springer, Nueva York.
- Gray, J. y Lampert, L. (2006). Consenso sobre la confirmación de transacciones. *ACM Trans. Database Syst.*, 31(1):133–160.
- Gray, J. y Reuter, A. (1993). *Procesamiento de transacciones: conceptos y técnicas*. Morgan Kaufmann.
- Gray, J., Helland, P., O'Neil, PE, y Shasha, D. (1996). Los peligros de la replicación y una solución. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*, páginas 173–182.
- Gray, JN, McJones, P., Blasgen, M., Lindsay, B., Lorie, R., Price, T., Putzolu, F. y Traiger, I. (1981). El gestor de recuperación del gestor de bases de datos System R. *ACM Comput. Surv.*, 13(2): 223–242.
- Grefen, P. y Widom, J. (1997). Protocolos para la comprobación de restricciones de integridad en bases de datos federadas. *Distribuidor Paralelo. Bases de datos*, 5(4):327–355.
- Griffiths, PP y Wade, BW (1976). Un mecanismo de autorización para una base de datos relacional. *ACM Trans. Database Syst.*, 1(3):242–255.
- Grossman, RL y Gu, Y. (2009). Sobre las variedades de nubes para computación intensiva en datos. *Q. Bull. IEEE TC sobre ingeniería de datos*, 32(1):44–50.
- Guha, S. y McGregor, A. (2006). Cuantiles aproximados y orden de la corriente. En *Proc. Simposio ACM SIGACT-SIGMOD sobre principios de sistemas de bases de datos*, páginas 273–279.
- Gulisano, V., Jiménez-Peris, R., Patino-Martínez, M. y Valdúriez, P. (2010). StreamCloud: Un sistema de transmisión de datos a gran escala. En *Actas de la 30.ª Conferencia Internacional del IEEE sobre Sistemas de Computación Distribuida*.
- Gulisano, V., Jiménez-Peris, R., Patino-Martínez, M. y Valdúriez, P. (2012). StreamCloud: Un sistema de transmisión de datos elástico y escalable. *IEEE Trans. Parall. Dist. Sys.*, 23(12):2351–2365.
- Gulli, A. y Signorini, A. (2005). La web indexable tiene más de 11.500 millones de páginas. En *Proc. 14ª Conferencia Internacional sobre la World Wide Web*, páginas 902–903.
- Gummadi, PK, Gummadi, R., Gribble, SD, Ratnasamy, S., Shenker, S. y Stoica, I. (2003). El impacto de la geometría de enrutamiento DHT en la resiliencia y la proximidad. En *Proc. Conf. sobre Aplicaciones, Tecnologías, Arquitecturas y Protocolos para la Comunicación Informática*, páginas 381–394.
- Güntzer, U., Kießling, W., y Balke, W.-T. (2000). Optimización de consultas multifunciones para bases de datos de imágenes. En *Actas de la 26.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 419–428.
- Gupta, A. y Mumick, IS, editores. (1999). *Vistas materializadas: técnicas, implementaciones, y aplicaciones*. MIT Press.
- Gupta, A., Mumick, IS, y Subrahmanian, VS (1993). Mantenimiento de vistas de forma incremental. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*, páginas 157–166.
- Gupta, A., Jagadish, H. y Mumick, IS (1996). Integración de datos mediante vistas automantenibles. En *Avances en tecnología de bases de datos, Actas de la 5.ª Conferencia Internacional sobre la extensión de la tecnología de bases de datos*, páginas 140–144.
- Gupta, A., Agrawal, D. y El Abbadi, A. (2003). Consultas de selección de rango aproximado en sistemas peer-to-peer. En *Actas de la 1.ª Conferencia Bienal sobre Investigación en Sistemas de Datos Innovadores*, págs. 141–151.
- Haas, L. (2007). *La Bella y la Bestia: Teoría y práctica de la integración de la información*. En *Proc. 11ª Conferencia Internacional sobre Teoría de Bases de Datos*, páginas 28–43.
- Haas, L., Kossmann, D., Wimmers, E. y Yang, J. (1997a). Optimización de consultas en diversas fuentes de datos. En *Actas de la 23.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 276–285.
- Haas, LM, Kossmann, D., Wimmers, EL, y Yang, J. (1997b). Optimización de consultas en diversas fuentes de datos. En *Actas de la 23.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 276–285.



- Haas, P. y Hellerstein, J. (1999a). Uniones de ondulación para agregación en línea. En Proc. ACM SIGMOD Conferencia Internacional sobre Gestión de Datos, páginas 287–298.
- Haas, PJ y Hellerstein, JM (1999b). Uniones de ondulación para agregación en línea. En Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos, páginas 287-298.
- Hacigümüs, H., Sankaranarayanan, J., Tatemura, J., LeFevre, J. y Polyzotis, N. (2013). Odyssey: Un sistema multialmacén para el análisis evolutivo. Proc. VLDB Endowment, 6(11): 1180–1181.
- Haderle, CMD, Lindsay, B., Pirahesh, H. y Schwarz, P. (1992). Aries: Un método de recuperación de transacciones que admite bloques de granularidad fina y reversiones parciales mediante registro de escritura anticipada. ACM Trans. Base de datos Syst., 17(1):94–162.
- Hadzilacos, V. (1988). Una teoría de la confiabilidad en sistemas de bases de datos. J. ACM, 35(1):121–145.
- Halevy, A., Rajaraman, A., y Ordille, J. (2006). Integración de datos: la adolescencia. En Proc. 32ª Conferencia Internacional sobre Bases de Datos Muy Grandes, páginas 9–16.
- Halevy, AY (2001). Respondiendo consultas mediante vistas: Una encuesta. VLDB J., 10(4):270–294.
- Halevy, AY, Etzioni, O., Doan, A., Ives, ZG, Madhavan, J., McDowell, L. y Tatarinov, I. (2003). Cruzando el abismo estructural. En Actas de la 1.ª Conferencia Bienal sobre Investigación Innovadora en Sistemas de Datos.
- Halici, U. y Dogac, A. (1989). Control de concurrencia en bases de datos distribuidas a través del tiempo. Intervalos y bloqueos a corto plazo. IEEE Trans. Softw. Eng., 15(8):994–995.
- Halim, F., Idreos, S., Karras, P. y Yap, RHC (2012). Cracking estocástico de bases de datos: Hacia una indexación adaptativa robusta en almacenes de columnas de memoria principal. Proc. VLDB Endowment, 5(6):502–513.
- Hammad, M., Aref, W. y Elmagarmid, A. (2003a). Unión de ventanas de flujo: Seguimiento de objetos en movimiento en bases de datos de redes de sensores. En Actas de la 15.ª Conferencia Internacional sobre Gestión de Bases de Datos Científicas y Estadísticas, páginas 75–84.
- Hammad, M., Aref, W., Franklin, M., Mokbel, M. y Elmagarmid, A. (2003b). Ejecución eficiente de consultas de ventana deslizante sobre flujos de datos. Informe técnico CSD TR 03-035, Universidad de Purdue.
- Hammad, M., Mokbel, M., Ali, M., Aref, W., Catlin, A., Elmagarmid, A., Eltabakh, M., Elfeke, M., Ghanem, T., Gwadera, R., Ilyas, I., Marzouk, M. y Xiong, X. (2004). Nile: un motor de procesamiento de consultas para flujos de datos. En Actas de la 20.ª Conferencia Internacional sobre Ingeniería de Datos, pág. 851.
- Hammad, M., Aref, W. y Elmagarmid, A. (2005). Optimización de la ejecución en orden de consultas continuas sobre datos de sensores transmitidos. En Actas de la 17.ª Conferencia Internacional sobre Gestión de Bases de Datos Científicas y Estadísticas, páginas 143-146.
- Hammer, M. y Niamir, B. (1979). Un enfoque heurístico para la partición de atributos. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 93-101.
- Hammer, M. y Shipman, DW (1980). Mecanismos de confiabilidad para SDD-1: Un sistema para bases de datos distribuidas. ACM Trans. Database Syst., 5(4):431–466.
- Han, M. (2015). Sobre la mejora de sistemas distribuidos de procesamiento de grafos tipo Pregel. Tesis de maestría. Universidad de Waterloo, Escuela de Ciencias de la Computación David R. Cheriton.
- Han, M. y Daudjee, K. (2015). Giraph Unchained: Ejecución paralela asíncrona sin barreras en sistemas de procesamiento de grafos tipo Pregel. Proc. VLDB Endowment, 8(9):950–961.
- Härder, T. y Reuter, A. (1983). Principios de recuperación de bases de datos orientadas a transacciones. ACM Comput. Surv., 15(4):287–317.
- Hartig, O. (2012). SPARQL para una web de datos enlazados: Semántica y computabilidad. En Proc. 9.ª Conferencia de Web Semántica Extendida, páginas 8-23.
- Hartig, O. (2013a). Una visión general de las estrategias de ejecución para consultas de datos vinculados. Datenbank-Spektrum, 13(2):89–99.
- Hartig, O. (2013b). SQUIN: un sistema de ejecución de consultas basado en recorrido para la web de datos vinculados. En Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos, páginas 1081–1084.
- Harvey, NJA, Jones, MB, Saroiu, S., Theimer, M. y Wolman, A. (2003). SkipNet: Una red superpuesta escalable con propiedades de localidad prácticas. En Actas del 4.º Simposio USENIX sobre Tecnología y Sistemas de Internet.

- He, B., Chang, KC-C., y Han, J. (2004). Minería de coincidencias complejas en interfaces de consulta web. En Proc. Taller ACM SIGMOD sobre temas de investigación en minería de datos y descubrimiento de conocimiento, páginas 3-10.
- He, Q. y Ling, T. W. (2006). Un enfoque basado en ontologías para la integración de entidades-relaciones. esquemas. *Data & Knowl. Eng.*, 58(3):299–326.
- Hedley, YL, Younas, M., James, A. y Sanderson, M. (2004a). Una técnica de muestreo de dos fases para la extracción de información de bases de datos web ocultas. En WIDM04, páginas 1–8.
- Hedley, Y.-L., Younas, M., James, AE, y Sanderson, M. (2004b). Extracción de datos de documentos web ocultos mediante consultas. En Actas de la 27.ª Conferencia Anual Internacional ACM SIGIR sobre Investigación y Desarrollo en Recuperación de Información, páginas 558-559.
- Heinze, T., Pappalardo, V., Jerzak, Z. y Fetzner, C. (2014). Técnicas de autoescalado para el procesamiento elástico de flujos de datos. En Actas de la 8.ª Conferencia Internacional sobre Sistemas Distribuidos Basados en Eventos, páginas 318-321.
- Heinze, T., Roediger, L., Meister, A., Ji, Y., Jerzak, Z. y Fetzner, C. (2015). Optimización de parámetros en línea para el procesamiento elástico de flujos de datos. En Actas del 6.º Simposio ACM sobre Computación en la Nube, páginas 276-287.
- Helal, AA, Heddaya, AA, y Bhargava, BB (1997). Técnicas de replicación en sistemas distribuidos. Kluwer Academic Publishers.
- Hellerstein, J. M., Haas, P. y Wang, H. (1997). Agregación en línea. En Proc. ACM SIGMOD. Conferencia Internacional sobre Gestión de Datos, páginas 171–182.
- Hellerstein, J. M., Franklin, M. J., Chandrasekaran, S., Deshpande, A., Hildrum, K., Madden, S., Raman, V. y Shah, MA (2000). Procesamiento adaptativo de consultas: Tecnología en evolución. Q. Bull. IEEE TC sobre ingeniería de datos, 23(2):7–18.
- Herlihy, M. (1987). Concurrencia versus disponibilidad: Mecanismos de atomicidad para datos replicados. *ACM Trans. Comp. Syst.*, 5(3):249–274.
- Herman, D. y Verjus, JP (1979). Un algoritmo para mantener la consistencia de múltiples copias. En Actas de la 1.ª Conferencia Internacional del IEEE sobre Sistemas de Computación Distribuida, páginas 625–631.
- Hersh, W. (2001). Gestión de gigabytes: compresión e indexación de documentos e imágenes (segundo edición). *Inf. Retr.*, 4(1):79–80.
- Hevner, AR y Schneider, GM (1980). Un sistema de diseño integrado para redes de bases de datos distribuidas. En Compendio de Artículos - COMPCON, páginas 459–465.
- Hirate, Y., Kato, S. y Yamana, H. (2006). Estructura web en 2005. En Actas del 4º Taller Internacional sobre Algoritmos y modelos para la Web-Graph, páginas 36 – 46.
- Hoffer, HA y Severance, DG (1975). El uso del análisis de conglomerados en el diseño de bases de datos físicas. En Proc. 1st Int. Conf. sobre bases de datos Very, páginas 69–86.
- Hoffer, JA (1975). Un enfoque de agrupamiento para la generación de subarchivos para el diseño de una base de datos informática. Tesis doctoral, Departamento de Investigación de Operaciones, Universidad de Cornell, Ithaca, NY.
- Hoffman, JL (1977). Métodos modelo para la seguridad informática y la privacidad. Prentice-Hall.
- Holze, M. y Ritter, N. (2008). Bases de datos autónomas: Detección de cambios en la carga de trabajo con modelos de n-gramas. En Proc. 12.ª Conferencia de Europa del Este. Avances en Bases de Datos y Sistemas de Información, páginas 127–142.
- Hong, W. (1992). Explotación del paralelismo entre operaciones en XPRS. En Proc. ACM SIGMOD Int. Conferencia sobre Gestión de Datos, páginas 19–28.
- Hong, W. y Stonebraker, M. (1993). Optimización de planes de ejecución de consultas paralelas en XPRS. Distribuidor Paralelo de Bases de Datos, 1(1):9–32.
- Hoque, I. y Gupta, I. (2013). LFGGraph: análisis de grafos distribuidos simple y rápido. En Proc. 1.ª edición. Conferencia ACM SIGOPS sobre resultados oportunos en sistemas operativos, páginas 9:1–9:17.
- Hortonworks. (2014). Libro blanco: Una arquitectura de datos moderna con Apache Hadoop: el camino hacia el lago de datos. Informe técnico, Hortonworks. Último acceso: agosto de 2018.
- Hsiao, HI y DeWitt, D. (1991). Estudio del rendimiento de tres estrategias de replicación de datos de alta disponibilidad. En Proc. Int. Conf. on Parallel and Distributed Information Systems, páginas 18-28.
- Huang, Z. y He, Y. (2018). Autodetección: Detección de errores basada en datos en tablas. En Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos, páginas 1377–1392.

- Huebsch, R., Hellerstein, J., Lanham, N., Loo, BT, Shenker, S. y Stoica, I. (2003). Consultas en internet con pier. En *Actas de la 29.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 321–332.
- Hull, R. (1997). Gestión de la heterogeneidad semántica en bases de datos: Una perspectiva teórica. En *Proc. Simposio ACM SIGACT-SIGMOD sobre principios de sistemas de bases de datos*, páginas 51–61.
- Hwang, J., Balazinska, M., Rasin, A., Cetintemel, U., Stonebraker, M. y Zdonik, S. (2005). Algoritmos de alta disponibilidad para el procesamiento distribuido de flujos. En *Actas de la 21.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 779–790.
- Idreos, S. (2010). *Cracking de bases de datos: Hacia el autoajuste de núcleos de bases de datos*. Tesis doctoral. Universidad de Ámsterdam.
- Idreos, S., Kersten, ML, y Manegold, S. (2007a). Actualización de una base de datos descifrada. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*, páginas 413–424.
- Idreos, S., Kersten, ML, y Manegold, S. (2007b). Cracking de bases de datos. En *Actas de la 3.ª Conferencia Bienal sobre Investigación en Sistemas de Datos Innovadores*, págs. 68–78.
- Idreos, S., Kersten, ML, y Manegold, S. (2009). Reconstrucción de tuplas autoorganizadas en almacenes de columnas. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 297–308.
- Idreos, S., Alagiannis, I., Johnson, R. y Ailamaki, A. (2011). Aquí están mis archivos de datos. Aquí están mis consultas. ¿Dónde están mis resultados? En *Actas de la 5.ª Conferencia Bienal sobre Investigación en Sistemas de Datos Innovadores*, páginas 57–68.
- Ilyas, I. y Chu, X. (2019). *Principios de limpieza de datos*. ACM Books.
- Ilyas, IF y Chu, X. (2015). Tendencias en la limpieza de datos relacionales: Consistencia y deduplicación. *Fundamentos y tendencias en bases de datos*, 5(4):281–393.
- Ilyas, IF, Beskales, G., y Soliman, MA (2008). Un estudio de técnicas de procesamiento de consultas top-k en sistemas de bases de datos relacionales. *ACM Comput. Surv.*, 40(4):1–58.
- Ioannidis, Y. y Wong, E. (1987). Optimización de consultas mediante recocido simulado. En *Proc. ACM. Conferencia Internacional SIGMOD sobre Gestión de Datos*, páginas 9–22.
- Ipeirotis, PG y Gravano, L. (2002). Búsqueda distribuida en la web oculta: Muestreo y selección de bases de datos jerárquicas. En *Actas de la 28.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 394–405.
- Irani, KB y Khabbaz, NG (1982). Una metodología para el diseño de redes de comunicación y la distribución de datos en sistemas informáticos distribuidos. *IEEE Trans. Comput.*, C-31(5): 419–434.
- Isloor, S. y Marsland, T. (1980). El problema del interbloqueo: Una visión general. *Computer*, 13(9):58–78.
- Ito, J., Narula, N. y Ali, R. (2017). La cadena de bloques tendrá en el sistema financiero el mismo efecto que internet tuvo en los medios de comunicación. Disponible en <https://hbr.org/2017/03/the-blockchain-will-do-to-banks-and-law-firms-what-the-internet-did-to-media/>. Último acceso febrero de 2019.
- Jagadish, HV, Ooi, BC y Vu, QH (2005). BATON: Una estructura de árbol balanceada para redes peer-to-peer. En *Actas de la 31.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 661–672.
- Jagadish, HV, Ooi, BC, Tan, K.-L., Vu, QH y Zhang, R. (2006). Aceleración de la búsqueda en redes peer-to-peer con una estructura de árbol multidireccional. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 1–12.
- Jajodia, S. y Mutchler, D. (1987). Votación dinámica. En *Actas de la Conferencia Internacional ACM SIGMOD sobre Gestión de datos*, páginas 227–238.
- Jajodia, S. y Sandhu, RS (1991). Hacia un modelo de datos relacionales seguro multinivel. En *Proc. Conferencia internacional ACM SIGMOD sobre gestión de datos*, páginas 50–59.
- Jajodia, S., Atluri, V., Keefe, TF, McCollum, CD y Mukkamala, R. (2001). Procesamiento de transacciones de seguridad multinivel. *J. Seguridad informática*, 9(3):165–195.
- Jhingran, AD, Mattos, N., y Piraresh, H. (2002). Integración de la información: una agenda de investigación. *IBM Systems J.*, 41(4):555–562.
- Jiménez-Peris, R. y Patiño Martínez, M. (2011). Sistema y método para el procesamiento transaccional descentralizado, altamente escalable y de baja contención. Patente estadounidense 9.760.597 B2, patente europea 2780832.
- Jiménez-Peris, R., Patiño-Martínez, M., y Alonso, G. (2002). Recuperación paralela no intrusiva de datos replicados. En *Proc. XXI Simposio sobre Sistemas Distribuidos Confiables*, páginas 150–159.

- Jiménez-Peris, R., Patiño-Martínez, M., Kemme, B. y Alonso, G. (2002). Mejora de la escalabilidad de clústeres de bases de datos con tolerancia a fallos. En *Actas de la 22.ª Conferencia Internacional del IEEE sobre Sistemas de Computación Distribuida*, páginas 477–484.
- Jiménez-Peris, R., Patiño-Martínez, M., Alonso, G. y Kemme, B. (2003). ¿Son los quórumes una alternativa para la replicación de datos? *ACM Trans. Database Syst.*, 28(3):257–294.
- Johnson, T., Muthukrishnan, S., Shkapenyuk, V. y Spatscheck, O. (2005). Un mecanismo de latido y su aplicación en Gigascope. En *Actas de la 31.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 1079–1088.
- Johnson, T., Muthukrishnan, SM, Shkapenyuk, V. y Spatscheck, O. (2008). Particionamiento basado en consultas para la monitorización de flujos masivos de datos de red. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 1135–1146.
- Kaelbling, LP, Littman, ML y Moore, AP (1996). Aprendizaje por refuerzo: Una encuesta. *J. Razonamiento automático*, 4:237–285.
- Kalogeraki, V., Gunopulos, D. y Zeinalipour-Yazti, D. (2002). Un mecanismo de búsqueda local para redes peer-to-peer. En *Actas de la 11.ª Conferencia Internacional sobre Gestión de la Información y el Conocimiento*, páginas 300–307.
- Kambayashi, Y., Yoshikawa, M. y Yajima, S. (1982). Procesamiento de consultas para bases de datos distribuidas mediante semiuniones generalizadas. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 151-160.
- Kang, J., Naughton, J. y Viglas, S. (2003). Evaluación de uniones de ventanas sobre flujos ilimitados. En *Proc. 19ª Conf. Int. sobre Ingeniería de Datos*, páginas 341–352.
- Kaoudi, Z. y Manolescu, I. (2015). RDF en las nubes: Un estudio. *VLDB J.*, 24:67–91.
- Kara, A., Ngo, HQ, Nikolic, M., Olteanu, D. y Zhang, H. (2019). Conteo de triángulos bajo actualizaciones en el peor de los casos. En *Actas de la 22.ª Conferencia Internacional sobre Teoría de Bases de Datos*, páginas 1:1–1:18.
- Karlalalem, K. y Navathe, SB (1994). Materialización de bases de datos relacionales distribuidas rediseñadas. Informe Técnico HKUST-CS94-14, Universidad de Ciencia y Tecnología de Hong Kong, Departamento de Informática.
- Karlalalem, K., Navathe, SB, y Ammar, M. (1996). Políticas óptimas de rediseño para el procesamiento dinámico de aplicaciones en un sistema de bases de datos relacionales distribuidas. *Inf. Syst.*, 21 (4):353–367.
- Karypis, G. y Kumar, V. (1995). Esquemas de partición de grafos multinivel. En *Proc. 1995, Conferencia Internacional sobre Procesamiento Paralelo*, páginas 113-122.
- Kashyap, V. y Sheth, AP (1996). Similitudes semánticas y esquemáticas entre objetos de bases de datos: Un enfoque basado en el contexto. *VLDB J.*, 5(4):276–304.
- Katz, B. y Lin, J. (2002). Anotación de la World Wide Web mediante lenguaje natural. En *Proc. 2.ª Taller sobre PLN y XML*, páginas 1–8.
- Kazerouni, L. y Karlalalem, K. (1997). Rediseño gradual de bases de datos relacionales distribuidas. Informe técnico HKUST-CS97-12, Universidad de Ciencia y Tecnología de Hong Kong, Departamento de Ciencias de la Computación.
- Keeton, K., Patterson, D., y Hellerstein, J. M. (1998). Un caso para discos inteligentes (idisks). *ACM SIGMOD Rec.*, 27(3):42–52.
- Keller, AM (1982). Actualización de bases de datos relacionales mediante vistas que incluyen uniones. En *Actas de la 2.ª Conferencia Internacional sobre Bases de Datos: Mejora de la Usabilidad y la Capacidad de Respuesta*, páginas 363–384.
- Kementsietsidis, A., Arenas, M., y Miller, RJ (2003). Gestión de mapeos de datos en el proyecto Hyperion. En *Actas de la 19.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 732–734.
- Kemme, B. y Alonso, G. (2000a). Un nuevo enfoque para el desarrollo e implementación de estrategias de comunicación entusiastas. *Protocolos de replicación de bases de datos. ACM Trans. Database Syst.*, 25(3):333–379.
- Kemme, B. y Alonso, G. (2000b). No sea perezoso, sea consistente: Postgres-R, una nueva forma de implementar la replicación de bases de datos. En *Proc. 26.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 134-143.
- Kemme, B., Bartoli, A. y Babaoglu, O. (2001). Reconfiguración en línea en bases de datos replicadas basada en comunicación grupal. En *Proc. Int. Conf. on Dependable Systems and Networks*, páginas 117-130.

- Kemme, B., Peris, R.J. y Patino-Martínez, M. (2010). Replicación de bases de datos. Morgan & Claypool.
- Kemper, A. y Neumann, T. (2011). HyPer: Un sistema híbrido de base de datos OLTP y OLAP en memoria principal basado en instantáneas de memoria virtual. En *Actas de la 27.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 195-206.
- Kermerrec, A.-M. y van Steen, M. (2007). Cotilleo en sistemas distribuidos. *Sistemas operativos*. Revista de Filosofía, 41(5):2-7.
- Kermerrec, A.-M., Rowstron, A., Shapiro, M. y Druschel, P. (2001). El enfoque del cubo de hielo para la reconciliación de réplicas divergentes. En *Proc. ACM SIGACT-SIGOPS 20.º Simposio sobre los Principios de la Computación Distribuida*, páginas 210-218.
- Khayyat, Z., Awara, K., Alonazi, A., Jamjoom, H., Williams, D. y Kalnis, P. (2013). Mizan: Un sistema para el balanceo dinámico de carga en el procesamiento de grafos a gran escala. En *Actas de la 8.ª Conferencia Europea ACM SIGOPS/EuroSys sobre Sistemas de Computación*, páginas 169-182.
- Khoshafian, S. y Valduriez, P. (1987). Persistencia compartida y orientación a objetos: Una perspectiva de bases de datos. En el *Taller Internacional sobre Lenguajes de Programación de Bases de Datos*, páginas 181-205.
- Kim, W. y Seo, J. (1991). Clasificación de la heterogeneidad esquemática y de datos en múltiples bases de datos. *sistemas. Computadora*, 24(12):12-18.
- Kirsch, J. y Amir, Y. (2008). Paxos para desarrolladores de sistemas: Una visión general. En *Actas del 2.º Taller sobre Sistemas Distribuidos a Gran Escala y Middleware*, páginas 3:1-3:6.
- Kitsuregawa, M. y Ogawa, Y. (1990). Hash paralelo con propagación de cubos: Un nuevo y robusto método de unión de hash paralelo para la asimetría de datos en la supercomputadora de bases de datos. En *Actas de la 16.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 210-221.
- Kitsuregawa, M., Tanaka, H. y Moto-Oka, T. (1983). Aplicación del hash a la máquina de base de datos y su arquitectura. *Computación de Nueva Generación*, 1(1):63-74.
- Kiveris, R., Lattanzi, S., Mirrokni, V., Rastogi, V. y Vassilvitskii, S. (2014). Componentes conectados en MapReduce y más allá. En *Actas del 5.º Simposio ACM sobre Computación en la Nube*, páginas 18:1-18:13.
- Kleinberg, J., Kumar, R., Raghavan, P., Rajagopalan, S. y Tomkins, A. (1999). La web como grafo: Medidas, modelos y métodos. En *Actas de la 5.ª Conferencia Internacional Anual sobre Computación y Combinatoria*, páginas 1-17.
- Kleinberg, JM (1999). Fuentes autorizadas en un entorno hipervinculado. *J. ACM*, 46(5): 604-632.
- Knapp, E. (1987). Detección de interbloqueos en bases de datos distribuidas. *ACM Comput. Surv.*, 19(4):303-328.
- Knuth, DE (1973). *El arte de la programación informática, Volumen III: Ordenamiento y búsqueda*. Addison-Wesley.
- Koch, C. (2001). Integración de datos frente a múltiples esquemas autónomos evolutivos. Tesis doctoral. Universidad Técnica de Viena.
- Koch, C. (2010). Evaluación incremental de consultas en un anillo de bases de datos. En *Actas del 29.º Congreso ACM SIGACT. Simposio SIGMOD-SIGART sobre principios de sistemas de bases de datos*, páginas 87-98.
- Koch, C., Ahmad, Y., Kennedy, O., Nikolic, M., Nötzli, A., Lupei, D. y Shaikhha, A. (2014). DBToaster: procesamiento delta de orden superior para vistas dinámicas y frecuentemente actualizadas. *VLDB J.*, 23(2): 253-278.
- Kohler, WH (1981). Un estudio de técnicas de sincronización y recuperación en sistemas descentralizados. *sistemas informáticos. ACM Comput. Surv.*, 13(2):149-183.
- Kolev, B., Bondiombouy, C., Valduriez, P., Jiménez-Peris, R., Pau, R. y Pereira, J. (2016a). El sistema multitienda cloudmssql. En *Proc. ACM SIGMOD Int. Conf. sobre gestión de datos*, páginas 2113-2116.
- Kolev, B., Valduriez, P., Bondiombouy, C., Jiménez-Peris, R., Pau, R. y Pereira, J. (2016b). CloudMdsQL: consulta de almacenes de datos heterogéneos en la nube con un lenguaje común. *Distribuido. Paralelo. Bases de datos*, 34(4):463-503.
- Kolev, B., Levchenko, O., Pacitti, E., Valduriez, P., Vilaça, R., Gonçalves, RC, Jiménez-Peris, R. y Kranas, P. (2018). Procesamiento paralelo de consultas políglotas en almacenes de datos heterogéneos en la nube con LeanXcale. En *Proc. 2018 IEEE Int. Conf. sobre Big Data*, páginas 1757-1766.

- Koller, D. y Friedman, N. (2009). Modelos gráficos probabilísticos: principios y técnicas. La prensa del MIT.
- Kollias, JG y Hatzopoulos, M. (1981). Criterios para facilitar la resolución del problema de asignación de copias de un archivo en una red informática. *Comp. J.*, 24(1):29–30.
- Konopnicki, D. y Shmueli, O. (1995). W3QS: Un sistema de consulta para la World Wide Web. En *Actas de la 21.ª Conferencia Internacional sobre Bases de Datos de Gran Tamaño*, páginas 54-65.
- Kossmann, D. (2000). El estado del arte en el procesamiento distribuido de consultas. *ACM Comput. Surv.*, 32(4):422–469.
- Krishnamurthy, R., Litwin, W. y Kent, W. (1991). Características del lenguaje para la interoperabilidad de bases de datos con discrepancias esquemáticas. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 40-49.
- Kshemkalyani, A. y Singhal, M. (1994). Sobre la caracterización y corrección de la distribución puntos muertos. *J. Parall. y Distrib. Comput.*, 22(1):44–59.
- Kubiatowicz, J., Bindel, D., Chen, Y., Czerwinski, S., Eaton, P., Geels, D., Gummadi, R., Rhea, S., Weatherspoon, H., Weimer, W., Wells, C. y Zhao, B. (2000). Oceanstore: una arquitectura para almacenamiento persistente a escala global. En la *Conferencia Internacional ACM sobre Soporte Arquitectónico para Lenguajes de Programación y Sistemas Operativos (ASPLOS)*, páginas 190–201.
- Kulkarni, S., Bhagat, N., Fu, M., Kedigehalli, V., Kellogg, C., Mittal, S., Patel, J. M., Ramasamy, K. y Taneja, S. (2015). Garza de Twitter: Procesamiento de flujo a escala. En *Proc. Conferencia Internacional ACM SIGMOD sobre Gestión de Datos*, páginas 239-250.
- Kumar, A. y Segev, A. (1993). Intercambios de costo y disponibilidad en la concurrencia de datos replicados. *control. ACM Trans. Database Syst.*, 18(1):102–131.
- Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A. y Upfal, E. (2000). La Web como grafo. En *Actas del XIX Simposio ACM SIGACT-SIGMOD-SIGART sobre Principios de Sistemas de Bases de Datos*, páginas 1-10.
- Kumar, V., editor. (1996). Rendimiento de los mecanismos de control de concurrencia en sistemas centralizados. *Sistemas de bases de datos*. Prentice-Hall.
- Kung, H. y Robinson, J. (1981). Sobre métodos optimistas para el control de concurrencia. *ACM Trans. Base de datos Syst.*, 6(2):213–226.
- Kwok, CCT, Etzioni, O., y Weld, DS (2001). Adaptación de la respuesta a preguntas a la web. En *Actas de la 10.ª Conferencia Internacional sobre la World Wide Web*, páginas 150-161.
- Ladwig, G. y Tran, T. (2010). Estrategias de procesamiento de consultas de datos enlazados. En *Proc. 9.ª Conferencia Internacional de Web Semántica*, páginas 453–469.
- Ladwig, G. y Tran, T. (2011). SIHJoin: Consulta de datos enlazados remotos y locales. En *Proc. 8.ª Conferencia de Web Semántica Extendida*, páginas 139-153.
- Lage, JP, da Silva, AS, Golgher, PB y Laender, AHF (2002). Recopilación de páginas web ocultas para la extracción de datos. En *Actas del 4.º Taller Internacional sobre Información Web y Gestión de Datos*, páginas 69-75.
- Lakshmanan, LVS, Sadri, F., y Subramanian, IN (1996). Un lenguaje declarativo para consultar y reestructurar la web. En *Actas del 6.º Taller Internacional sobre Investigación en Ingeniería de Datos*, páginas 12-21.
- Lam, SS y Özsu, MT (2002). Consulta de datos web: el enfoque WebQA. En *Proc. 3.ª edición. Conferencia sobre Ingeniería de Sistemas de Información Web*, páginas 139–148.
- Lamport, L. (1998). El parlamento a tiempo parcial. *ACM Trans. Comp. Syst.*, 16(2):133–169.
- Lamport, L. (2001). Paxos simplificado. *ACM SIGACT News*, 32(4):51–58.
- Lampson, B. y Sturgis, H. (1976). Recuperación de fallos en sistemas de almacenamiento de datos distribuidos. Técnico. Informe, Centro de Investigación Xerox Palo Alto, Palo Alto, California.
- Landers, T. y Rosenberg, RL (1982). Una visión general de multibase. En Schneider, H.-J., editor, *Bases de Datos Distribuidas*, páginas 153-184. Holanda Septentrional, Ámsterdam.
- Langville, AN y Meyer, CD (2006). PageRank de Google y más allá. Universidad de Princeton. Prensa.
- Lanzelotte, R., Valduriez, P., Zaït, M. y Ziane, M. (1994). Optimización de consultas paralelas de alto rendimiento: problemas y lecciones. *Inf. Syst.*, 19(4):311–330.

- Larriba-Pey, JL, Martínez-Bazán, N. y Domínguez-Sal, D. (2014). Introducción a las bases de datos gráficas. En Koubarakis, M., Stamou, G., Stoilos, G., Horrocks, I., Kolaitis, P., Lausen, G. y Weikum, G., editores, *Reasoning Web: Reasoning on the Web in the Big Data Era*, páginas 171–194. Saltador.
- Larson, P.-Å., Blanas, S., Diaconu, C., Freedman, C., Patel, JM y Zwilling, M. (2011). Mecanismos de control de concurrencia de alto rendimiento para bases de datos en memoria principal. *Proc. VLDB Endowment*, 5(4):298–309.
- Law, Y.-N., Wang, H. y Zaniolo, C. (2004). Lenguajes de consulta y modelos de datos para secuencias y flujos de datos de bases de datos. En *Actas de la 30.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 492-503.
- Lawrence, S. y Giles, CL (1998). Búsqueda en la World Wide Web. *Science*, 280(5360):98–100.
- Lawrence, S. y Giles, CL (1999). Accesibilidad de la información en la web. *Nature*, 400(6740): 107–9.
- Lee, K.-H., Lee, Y.-J., Choi, H., Chung, YD y Moon, B. (2012). Procesamiento de datos paralelos con mapreduce: una encuesta. *ACM SIGMOD Rec.*, 40(4):11–20.
- LeFevre, J., Sankaranarayanan, J., Hacigumus, H., Tatemura, J., Polyzotis, N. y Carey, MJ (2014). MISO: Optimización del procesamiento de consultas de big data con un sistema multialmacén. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 1591-1602.
- Leis, V., Boncz, PA, Kemper, A. y Neumann, T. (2014). Paralelismo basado en Morsel: un marco de evaluación de consultas compatible con NUMA para la era multinúcleo. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 743–754.
- Lenoski, D., Laudon, J., Gharachorloo, K., Weber, WD, Gupta, A., Henessy, J., Horowitz, M. y Lam, MS (1992). El multiprocesador Stanford Dash. *Computer*, 25(3):63–79.
- Lenzerini, M. (2002). Integración de datos: una perspectiva teórica. En *Proc. ACM SIGACT-SIGMOD Simposio sobre principios de sistemas de bases de datos*, páginas 233–246.
- Levandowski, JJ, Larson, P. Å., y Stoica, R. (2013). Identificación de datos calientes y fríos en bases de datos de memoria principal. En *Actas de la 29.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 26-37.
- Levin, KD y Morgan, HL (1975). Optimización de bases de datos distribuidas: Un marco para la investigación. En *Proc. National Computer Conf*, páginas 473–478.
- Levy, AY, Mendelzon, AO, Sagiv, Y. y Srivastava, D. (1995). Respuesta a consultas mediante vistas. En *Proc. ACM SIGACT-SIGMOD Symp. on Principles of Database Systems*, páginas 95-104.
- Levy, AY, Rajaraman, A., y Ordille, JJ (1996a). La World Wide Web como una colección de vistas: Procesamiento de consultas en la variedad de información. En *Proc. Taller sobre Vistas Materializadas: Técnicas y Aplicaciones*, páginas 43–55.
- Levy, AY, Rajaraman, A., y Ordille, JJ (1996b). Consulta de fuentes de información heterogéneas mediante descripciones de fuentes. En *Actas de la 22.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 251-262.
- Li, F., Ooi, BC, Özsu, MT y Wu, S. (2014). Gestión de datos distribuidos mediante MapReduce. *ACM Comput. Surv.*, 46(3):Artículo N° 31.
- Li, H.-G., Chen, S., Tatemura, J., Agrawal, D., Candan, KS y Hsiung, W.-P. (2006). Garantía de seguridad de consultas de unión continua sobre flujos de datos puntuados. En *Actas de la 32.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 19-30.
- Li, J., Maier, D., Tufte, K., Papadimos, V. y Tucker, P. a. (2005). Semántica y técnicas de evaluación para agregados de ventana en flujos de datos. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 311–322.
- Li, W.-S. y Clifton, C. (2000). Semint: Una herramienta para identificar correspondencias de atributos en bases de datos heterogéneas mediante redes neuronales. *Data & Knowl. Eng.*, 33(1):49–84.
- Li, W.-S., Clifton, C., y Liu, S.-Y. (2000). Integración de bases de datos mediante redes neuronales: Implementación y experiencias. *Knowl. and Information Syst.*, 2(1):73–96.
- Lim, L., Wang, M., Padmanabhan, S., Vitter, JS y Agarwal, R. (2003). Mantenimiento dinámico de índices web mediante puntos de referencia. En *Actas de la 12.ª Conferencia Internacional sobre la World Wide Web*, páginas 102-111.
- Lima, A., Mattoso, M. y Valduriez, P. (2004). Procesamiento de consultas OLAP en un clúster de bases de datos. En *Proc. 10ª Int. Euro-Par Conf.*, páginas 355–362.

- Lin, Q., Chang, P., Chen, G., Ooi, BC, Tan, K. y Wang, Z. (2016). Hacia una gestión de transacciones no 2PC en sistemas de bases de datos distribuidas. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 1659–1674.
- Lin, Y., Kemme, B., Patiño Martínez, M., y Jiménez-Peris, R. (2005). Replicación de datos basada en middleware que proporciona aislamiento de instantáneas. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 419–430.
- Litwin, W., Neimat, M.-A. y Schneider, DA (1993). LH\*: hash lineal para archivos distribuidos. En Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos, páginas 327–336.
- Liu, B., Zhu, Y. y Rundensteiner, E. (2006). Desbordamiento de estado de operadores en tiempo de ejecución para consultas de larga duración con uso intensivo de memoria. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 347–358.
- Livny, M., Khoshafian, S. y Boral, H. (1987). Gestión multidisco. En Proc. ACM SIGMETRICS Conf. sobre Medición y Modelado de Sistemas Informáticos, páginas 69–77.
- Lohman, G., Mohan, C., Haas, L., Daniels, D., Lindsay, B., Selinger, P. y Wilms, P. (). Consulta procesamiento en R\*. páginas 31–47.
- Lomet, D., Feket, A., Wang, R. y Ward, P. (2012). Concurrencia multiversión mediante la gestión de conflictos de rangos de marcas de tiempo. En Actas de la 28.ª Conferencia Internacional sobre Ingeniería de Datos, páginas 714-725.
- Low, Y., Gonzalez, J., Kyrola, A., Bickson, D. y Guestrin, C. (2010). GraphLab: nuevo marco para el aprendizaje automático paralelo. En Actas de la 26.ª Conferencia sobre Incertidumbre en Inteligencia Artificial, páginas 340–349.
- Low, Y., Gonzalez, J., Kyrola, A., Bickson, D., Guestrin, C. y Hellerstein, JM (2012). Graphlab distribuido: Un marco para el aprendizaje automático en la nube. Proc. VLDB Endowment, 5(8):716–727.
- Lu, H., Shan, M.-C. y Tan, K.-L. (1991). Optimización de consultas de unión multidireccional para ejecución paralela. En Actas de la 17.ª Conferencia Internacional sobre Bases de Datos Muy Grandes, páginas 549-560.
- Lu, H., Ooi, B. y Goh, C. (1992). Optimización global de consultas multibase de datos. ACM SIGMOD Rec., 21(4):6–11.
- Lu, H., Ooi, B. y Goh, C. (1993). Optimización de consultas multibase de datos: Problemas y soluciones. En Actas del 3er Taller Internacional sobre Cuestiones de Res. en Ingeniería de Datos, páginas 137–143.
- Lugowski, A., Alber, D., Buluç, A., Gilbert, JR, Reinhardt, S., Teng, Y. y Waranis, A. (2012). Una caja de herramientas flexible y de código abierto para el análisis escalable de grafos complejos. En Proc. 2012 SIAM Int. Conferencia sobre Minería de Datos, páginas 930–941.
- Lumsdaine, A., Gregor, D., Hendrickson, B. y Berry, J. (2007). Desafíos en gráficos paralelos. procesamiento. Cartas de procesamiento paralelo, 17(01):5–20.
- Lunt, TF y Fernández, EB (1990). Seguridad de bases de datos. ACM SIGMOD Rec., 19(4):90–97.
- Lv, Q., Cao, P., Cohen, E., Li, K. y Shenker, S. (2002). Búsqueda y replicación en redes peer-to-peer no estructuradas. En Actas de la 16.ª Conferencia Internacional Anual sobre Supercomputación, páginas 84-95.
- Lynch, N. (1983a). Atomicidad multinivel: Un nuevo criterio de corrección para la concurrencia de bases de datos. control. ACM Trans. Database Syst., 8(4):484–502.
- Lynch, N. (1983b). Control de concurrencia para transacciones anidadas resilientes. En Actas del 2.º Simposio ACM SIGACT-SIGMOD sobre Principios de Sistemas de Bases de Datos, páginas 166-181.
- Lynch, N. y Merritt, M. (1986). Introducción a la teoría de transacciones anidadas. Informe técnico MIT/LCS/TR-367, Instituto Tecnológico de Massachusetts, Cambridge, Massachusetts.
- Lynch, N., Merritt, M., Weihl, WE, y Fekete, A. (1993). Transacciones atómicas en sistemas distribuidos concurrentes. Morgan Kaufmann.
- Mackert, L. y Lohman, G. (1986a). Validación del optimizador R\* y evaluación del rendimiento para consultas distribuidas. En Actas de la 12.ª Conferencia Internacional sobre Bases de Datos Muy Grandes, páginas 149-159.
- Mackert, LF y Lohman, G. (1986b). Validación del optimizador R\* y evaluación del rendimiento para consultas locales. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 84-95.
- Madden, S. y Franklin, MJ (2002). Fjording the stream: Una arquitectura para consultas sobre datos de sensores en streaming. En Actas de la 18.ª Conferencia Internacional sobre Ingeniería de Datos, páginas 555–566.
- Madden, S., Shah, M., Hellerstein, J. y Raman, V. (2002a). Consultas continuas adaptativas sobre flujos. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 49-60.



- Madden, S., Shah, MA, Hellerstein, JM y Raman, V. (2002b). Consultas continuas adaptativas sobre flujos. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 49-60.
- Madhavan, J., Bernstein, P. y Rahm, E. (2001). Coincidencia de esquemas genéricos con Cupido. En Proc. 27ª Conferencia Internacional sobre Bases de Datos Muy Grandes, páginas 49-58.
- Mahmoud, A. y Riordon, J. S. (1976). Asignación óptima de recursos en sistemas distribuidos. redes de información. ACM Trans. Database Syst., 1(1):66-78.
- Maiyya, S., Zakhary, V., Agrawal, D. y El Abbadi, A. (2018). Fundamentos de bases de datos y computación distribuida para el mantenimiento escalable, tolerante a fallos y consistente de cadenas de bloques. Proc. Dotación VLDB, 11(12):2098-2101.
- Malewicz, G., Austern, MH, Bik, AJC, Dehnert, JC, Horn, I., Leiser, N. y Czajkowski, G. (2010). Pregel: un sistema para el procesamiento de grafos a gran escala. En Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos, páginas 135-146.
- Manber, U. y Myers, G. (1990). Matrices de sufijos: un nuevo método para búsquedas de cadenas en línea. En Proc. 1.er Simposio Anual ACM-SIAM sobre Algoritmos Discretos, páginas 319-327.
- Manegold, S., Boncz, PA, y Kersten, ML (2002). Optimización de la unión a memoria principal en hardware moderno. IEEE Trans. Knowl. and Data Eng., 14(4):709-730.
- Manolescu, I., Florescu, D. y Kossmann, D. (2001). Respuesta a consultas XML en fuentes de datos heterogéneas. En Actas de la 27.ª Conferencia Internacional sobre Bases de Datos Muy Grandes, páginas 241-250.
- Martins, V. y Pacitti, E. (2006). Conciliación dinámica y distribuida en redes p2p-dht. En Conferencia Europea sobre Computación Paralela (Euro-Par), páginas 337-349.
- Martins, V., Akbarinia, R., Pacitti, E. y Valdúriez, P. (2006a). Reconciliación en el sistema P2P APPA. En Proc. IEEE Int. Conf. sobre Sistemas Paralelos y Distribuidos, páginas 401-410.
- Martins, V., Pacitti, E., y Valdúriez, P. (2006b). Estudio sobre la replicación de datos en sistemas P2P. Informe Técnico 6083, INRIA, Rennes, Francia.
- Martins, V., Pacitti, E., Dick, ME, y Jiménez-Peris, R. (2008). Escalable y con reconocimiento de topología. Reconciliación en redes P2P. Distribuidora Paralela. Bases de Datos, 24(1-3):1-43.
- McBrien, P. y Poulavassilis, A. (2003). Definición de la integración de datos peer-to-peer utilizando ambos como reglas de vista. En Proc. 1er Taller Internacional sobre Bases de Datos, Sistemas de Información y Computación Peer-to-Peer, páginas 91-107.
- McCallum, A., Nigam, K., Rennie, J. y Seymore, K. (1999). Un enfoque de aprendizaje automático para Desarrollo de motores de búsqueda específicos de dominio. En Actas de la 16.ª Conferencia Conjunta Internacional sobre IA.
- McCann, R., AlShebli, B., Le, Q., Nguyen, H., Vu, L. y Doan, A. (2005). Mantenimiento de mapas para sistemas de integración de datos. En Actas de la 31.ª Conferencia Internacional sobre Bases de Datos Muy Grandes, páginas 1018-1029.
- McCormick, WT, Schweitzer, PJ, y White, TW (1972). Descomposición de problemas y reorganización de datos mediante una técnica de agrupamiento. Oper. Res., 20(5):993-1009.
- McCune, RR, Weninger, T., y Madey, G. (2015). Pensando como un vértice: Un estudio de marcos centrados en vértices para el procesamiento distribuido de grafos a gran escala. ACM Comput. Surv., 48(2): 25:1-25:39.
- Mehta, M. y DeWitt, D. (1995). Gestión del paralelismo intraoperador en sistemas de bases de datos paralelas. En Actas de la 21.ª Conferencia Internacional sobre Bases de Datos de Gran Tamaño.
- Melnik, S., Raghavan, S., Yang, B. y García-Molina, H. (2001). Desarrollo de un sistema distribuido de texto completo. Índice para la web. En Actas de la 10.ª Conferencia Internacional sobre la World Wide Web, páginas 396-406.
- Melnik, S., García-Molina, H., y Rahm, E. (2002). Inundación de similitud: Un algoritmo versátil de coincidencia de grafos y su aplicación a la coincidencia de esquemas. En Actas de la 18.ª Conferencia Internacional sobre Ingeniería de Datos, páginas 117-128.
- Menasce, DA y Muntz, RR (1979). Detección de bloqueos y puntos muertos en bases de datos distribuidas. Traducción IEEE. Software. Ing., SE-5(3):195-202.
- Mendelzon, AO, Mihaila, GA, y Milo, T. (1997). Consultas en la World Wide Web. Int. J. Digit. Librería, 1(1):54-67.
- Meng, W., Yu, C., Kim, W., Wang, G., Phan, T. y Dao, S. (1993). Construcción de una interfaz relacional para sistemas de bases de datos orientados a objetos. En Actas de la 9.ª Conferencia Internacional sobre Ingeniería de Datos, páginas 476-483.

- Milán-Franco, JM, Jiménez-Peris, R., Patiño-Martínez, M. y Kemme, B. (2004). Middleware adaptativo para replicación de datos. En Proc. ACM/IFIP/USENIX 5th Int. Middleware Conf., páginas 175–194.
- Miller, RJ, Haas, LM y Hernández, MA (2000). Mapeo de esquemas como descubrimiento de consultas. En Actas de la 26.ª Conferencia Internacional sobre Bases de Datos Muy Grandes, páginas 77–88.
- Miller, RJ, Hernández, MA, Haas, LM, Yan, L., Ho, CTH, Fagin, R. y Popa, L. (2001). El proyecto Clio: Gestión de la heterogeneidad. ACM SIGMOD Rec., 31(1):78–83.
- Milo, T. y Zohar, S. (1998). Uso de la correspondencia de esquemas para simplificar la traducción de datos heterogéneos. En Proc. 24ª Conf. Int. sobre bases de datos muy grandes, páginas 122–133.
- Minoura, T. y Wiederhold, G. (1982). Esquema resiliente de tokens de copia verdadera extendidos para un Sistema de base de datos distribuida. IEEE Trans. Softw. Eng., SE-8(3):173–189.
- Mitchell, T. (1997). Aprendizaje automático. McGraw-Hill.
- Mitzenmacher, M. (2001). El poder de dos opciones en el balanceo de carga aleatorio. IEEE Trans. Parall. Dist. Sys., 12(10):1094–1104.
- Mohan, C. (1979). Diseño de bases de datos en entornos distribuidos. Documento de trabajo WP-7902. Departamento de Ciencias de la Computación, Universidad de Texas en Austin.
- Mohan, C. y Lindsay, B. (1983). Protocolos de confirmación eficientes para el modelo de árbol de procesos de transacciones distribuidas. En Proc. ACM SIGACT-SIGOPS 2.º Simposio sobre los Principios de la Computación Distribuida, páginas 76–88.
- Mohan, C. y Yeh, RT (1978). Sistemas de bases de datos distribuidas: Un marco para el diseño de bases de datos. En Bases de datos distribuidas, Informe de estado del arte de Infotech. Infotech.
- Mohan, C., Lindsay, B. y Obermarck, R. (1986). Gestión de transacciones en el sistema distribuido r\*. sistema de gestión de bases de datos. ACM Trans. Database Syst., 11(4):378–396.
- Morgan, HL y Levin, KD (1977). Localización óptima de programas y datos en redes informáticas. Comun. ACM, 20(5):315–322.
- Moss, E. (1985). Transacciones anidadas. MIT Press.
- Muthukrishnan, S. (2005). Flujos de datos: Algoritmos y aplicaciones. Fundamentos y tendencias. Licenciatura en Informática Teórica. NOW Publishers.
- Naacke, H., Tomasic, A., y Valduriez, P. (1999). Validación de modelos de costes de mediadores con DISCO. Revista de redes y sistemas de información, 2(5):639–663.
- Najork, M. y Wiener, J.L. (2001). El rastreo en amplitud produce páginas de alta calidad. En Proc. 10ª Conferencia Internacional sobre la World Wide Web, páginas 114–118.
- Nakamoto, S. (2008). Bitcoin: Un sistema de efectivo electrónico entre pares. Disponible en <https://bitcoin.org/bitcoin.pdf>. Último acceso febrero de 2019.
- Nasir, MAU, Morales, GDF, García-Soriano, D., Kourtellis, N., y Serafini, M. (2015). El poder de ambas opciones: Balanceo de carga práctico para motores de procesamiento de flujos distribuidos. En Actas de la 31.ª Conferencia Internacional sobre Ingeniería de Datos, páginas 137–148.
- Nasir, MAU, Morales, GDF, Kourtellis, N., y Serafini, M. (2016). Cuando dos opciones no son suficientes: Equilibrio a escala en el procesamiento de flujos distribuidos. En Actas de la 32.ª Conferencia Internacional sobre Ingeniería de Datos, páginas 589–600.
- Naumann, F., Ho, C.-T., Tian, X., Haas, LM y Megiddo, N. (2002). Clasificación de atributos mediante análisis de características. En Actas de la 18.ª Conferencia Internacional sobre Ingeniería de Datos, pág. 271.
- Navathe, SB, Ceri, S., Wiederhold, G. y Dou, J. (1984). Partición vertical de algoritmos. para el diseño de bases de datos. ACM Trans. Database Syst., 9(4):680–710.
- Nejdl, W., Siberski, W. y Sintek, M. (2003). Problemas y desafíos de diseño para RDF y esquemas. sistemas peer-to-peer basados en ACM. Rec. SIGMOD, 32(3):41–46.
- Nepal, S. y Ramakrishna, M. (1999). Problemas de procesamiento de consultas en bases de datos de imágenes (multimedia). En Proc. 15ª Conf. Int. sobre Ingeniería de Datos, páginas 22–29.
- Neumann, T. y Weikum, G. (2008). RDF-3X: un motor de estilo RISC para RDF. Proc. VLDB Endowment, 1(1):647–659.
- Neumann, T. y Weikum, G. (2009). El motor RDF-3X para la gestión escalable de datos RDF. VLDB J., 19(1):91–113.

- Newman, MEJ, Watts, DJ y Strogatz, SH (2002). Modelos de grafos aleatorios de redes sociales. En (Coloquio Sackler NAS) Complejidad autoorganizada en las ciencias físicas, biológicas y sociales, págs. 2566-2573. Academia Nacional de Ciencias.
- Niamir, B. (1978). Partición de atributos en un sistema de base de datos relacional autoadaptativo. Informe Técnico 192, Laboratorio de Ciencias de la Computación, Instituto Tecnológico de Massachusetts, Cambridge, Massachusetts.
- Nicolas, JM (1982). Lógica para mejorar la comprobación de integridad en bases de datos relacionales. *Acta Informática*, 18:227–253.
- Nikolic, M. y Olteanu, D. (2018). Mantenimiento de vistas incrementales con ventajas de factorización de triple bloqueo. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 365–380.
- Novakovic, S., Daglis, A., Bugnion, E., Falsafi, B. y Grot, B. (2014). Escalabilidad horizontal de NUMA. En *Soporte arquitectónico para lenguajes de programación y sistemas operativos, ASPLOS*, páginas 3-18.
- Obermack, R. (1982). Algoritmo distribuido de detección de interbloqueos. *ACM Trans. Database Syst.*, 7 (2):187–208.
- Okcan, A. y Riedewald, M. (2011). Procesamiento de uniones theta con MapReduce. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*, páginas 949-960.
- Olston, C., Reed, B., Srivastava, U., Kumar, R. y Tomkins, A. (2008). Pig Latin: un lenguaje no tan extraño para el procesamiento de datos. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 1099-1110.
- Ong, KW, Papakonstantinou, Y., y Vernoux, R. (2014). El modelo de datos semiestructurado y el lenguaje de consulta SQL++: Un estudio de las capacidades de bases de datos SQL-on-Hadoop, NoSQL y NewSQL. *CoRR/abs/1405.3631*.
- Ongaro, D. y Ousterhout, J. (2014). En busca de un algoritmo de consenso comprensible. En *Proc. Conferencia Técnica Anual USENIX 2014*, páginas 305–320.
- Ooi, B., Shu, Y., y Tan, K.-L. (2003). Intercambio de datos relacionales en sistemas de gestión de datos entre pares. *ACM SIGMOD Rec.*, 32(3):59–64.
- Ouksel, AM y Sheth, AP (1999). Interoperabilidad semántica en sistemas de información globales: Breve introducción al área de investigación y a la sección especial. *ACM SIGMOD Rec.*, 28(1): 5–12.
- Özsoyoglu, ZM y Zhou, N. (1987). Procesamiento distribuido de consultas en radiodifusión de área local. *Redes*. En *Actas de la 20.ª Conferencia Internacional de Hawái sobre Ciencias de Sistemas*, páginas 419-429.
- Özsu, MT (2016). Un estudio de sistemas de gestión de datos RDF. *Front. Comput. Sci.*, 10(3): 418–432.
- Pacaci, A. y Özsu, MT (2018). Partición de flujos con enfoque en la distribución para sistemas de procesamiento de flujos distribuidos. En *Actas del 5.º Taller ACM SIGMOD sobre Algoritmos y Sistemas para MapReduce y posteriores*, páginas 6:1–6:10.
- Pacitti, E. y Simon, E. (2000). Actualización de las estrategias de propagación para mejorar la frescura en la planta maestra perezosa. *bases de datos replicadas. VLDB J.*, 8(3-4):305–318.
- Pacitti, E., Simon, E., y de Melo, R. (1998). Mejora de la frescura de los datos en esquemas maestros perezosos. En *Proc. 18a Conferencia Internacional IEEE sobre Sistemas de Computación Distribuida*, páginas 164-171.
- Pacitti, E., Minet, P. y Simon, E. (1999). Algoritmos rápidos para mantener la consistencia de las réplicas en bases de datos replicadas con un maestro diferido. En *Actas de la 25.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 126-137.
- Pacitti, E., Coulon, C., Valduriez, P. y Özsu, MT (2005). Replicación preventiva en una base de datos. *clúster. Distribu. Paralelo. Bases de datos*, 18(3):223–251.
- Pacitti, E., Valduriez, P. y Mattoso, M. (2007). Gestión de datos de red: problemas abiertos y nuevas cuestiones. *Revista de Computación en Red*, 5(3):273–281.
- Pacitti, E., Akbarinia, R. y Dick, ME (2012). Técnicas P2P para aplicaciones descentralizadas. *Conferencias de síntesis sobre gestión de datos*. Morgan & Claypool Publishers.
- Page, L., Brin, S., Motwani, R. y Winograd, T. (1998). El sistema de clasificación de citas PageRank: Orden a la web. *Informe técnico*, Universidad de Stanford.
- Palopoli, L. (2003). Experiencias con DIKE, un sistema para el diseño de sistemas de información cooperativos y almacenes de datos. *Inf. Syst.*, 28(7):835–865.

- Palopoli, L., Saccà, D., y Ursino, D. (1998). Descubrimiento semántico semiautomático de propiedades de esquemas de bases de datos. En *Actas de la 2.ª Conferencia Internacional sobre Ingeniería de Bases de Datos y Aplicaciones*, páginas 244-253.
- Palopoli, L., Saccà, D., Terracina, G. y Ursino, D. (1999). Un marco unificado basado en grafos para derivar propiedades nominales entre esquemas, conflictos de tipos y similitudes entre clústeres de objetos. En *Proc. Conferencia Internacional sobre Sistemas de Infantería Cooperativa*, páginas 34-45.
- Palopoli, L., Saccà, D., Terracina, G. y Ursino, D. (2003). Técnicas uniformes para derivar similitudes de objetos y subesquemas en bases de datos heterogéneas. *IEEE Trans. Knowl. and Data Eng.*, 15(2):271-294.
- Papadimitriou, CH (1986). La teoría del control de concurrencia. Computer Science Press.
- Papakonstantinou, Y., Garcia-Molina, H., y Widom, J. (1995). Intercambio de objetos entre fuentes de información heterogéneas. En *Actas de la 11.ª Conferencia Internacional sobre Ingeniería de Datos*, págs. 251-260.
- Pape, CL, Gañarski, S., y Valduriez, P. (2004). Refresco: Mejora del rendimiento de las consultas mediante el control de frescura en un clúster de bases de datos. En *Proc. Confederated Int. Conf. DOA, CoopIS y ODBASE, Lecture Notes in Computer Science* 3290, páginas 174-193.
- Paris, JF (1986). Votación con testigos: Un esquema de consistencia para archivos replicados. En *Proc. 6.ª IEEE Int. Conf. sobre sistemas de computación distribuida*, páginas 606-612.
- Pasetto, D. y Akhriev, A. (2011). Un estudio comparativo de algoritmos de ordenamiento paralelo. En *Proc. 26.ª Conferencia ACM SIGPLAN sobre Sistemas, Lenguajes y Aplicaciones de Programación Orientada a Objetos*, páginas 203-204.
- Passerini, A., Frasconi, P. y Soda, G. (2001). Métodos de evaluación para el rastreo enfocado. En *Actas del 7.º Congreso de la Asociación Italiana de Inteligencia Artificial*, págs. 33-39.
- Pasupuleti, P. y Purra, BS (2015). Desarrollo de lagos de datos con big data. Packt Books.
- Patiño-Martínez, M., Jiménez-Peris, R., Kemme, B. y Alonso, G. (2000). Replicación escalable en clústeres de bases de datos. En *Actas del 14.º Simposio Internacional sobre Computación Distribuida*, páginas 315-329.
- Pavlo, A., Paulson, E., Rasin, A., Abadi, DJ, DeWitt, DJ, Madden, S. y Stonebraker, M. (2009). Comparación de enfoques para el análisis de datos a gran escala. En *Proc. ACM SIGMOD Int. Conferencia sobre Gestión de Datos*, páginas 165-178.
- Pavlo, A., Curino, C. y Zdonik, SB (2012). Particionamiento automático de bases de datos con reconocimiento de sesgos en sistemas OLTP paralelos sin recursos compartidos. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 61-72.
- Perez-Sorrosal, F., Vuckovic, J., Patiño-Martínez, M., y Jiménez-Peris, R. (2006). Transacciones y actividades de larga duración y alta disponibilidad para J2EE. En *Actas de la 26.ª Conferencia Internacional del IEEE sobre Sistemas de Computación Distribuida*, página 2.
- Petraki, E., Idreos, S. y Manegold, S. (2015). Indexación holística en almacenes de columnas de memoria principal. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*, páginas 1153-1166.
- Pike, R., Dorward, S., Griesemer, R. y Quinlan, S. (2005). Interpretación de los datos: Análisis paralelo con sierra de vaivén. *Sci. Program.*, 13(4):277-298.
- Pirahesh, H., Mohan, C., Cheng, JM, Liu, TS y Selinger, PG (1990). Paralelismo en RDBMS: Problemas de arquitectura y diseño. En *Actas del 2.º Simposio Internacional sobre Bases de Datos en Sistemas Distribuidos y Paralelos*, páginas 4-29.
- Pirk, H., Manegold, S., y Kersten, M. (2014). Desperdicio no ... Coprocesamiento eficiente de datos relacionales. En *Actas de la 30.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 508-519.
- Plattner, C. y Alonso, G. (2004). Ganymed: Replicación escalable para aplicaciones web transaccionales. En *Proc. ACM/IFIP/USENIX 5.ª Conferencia Internacional de Middleware*, páginas 155-174.
- Plugge, E., Membrey, P., y Hawkins, T. (2010). La guía definitiva de MongoDB: NoSQL Base de datos para computación en la nube y de escritorio. Apress.
- Popa, L., Velegrakis, Y., Miller, RJ, Hernandez, MA, y Fagin, R. (2002). Traduciendo la web datos. En *Actas de la 28.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*.
- Porto, F., Laber, ES, y Valduriez, P. (2003). Selección selectiva: Una estrategia de procesamiento de consultas semánticas para la evaluación de predicados costosos. En *Proc. Simposio Brasileño de Bases de Datos*, páginas 356-370.

- Ports, DRK y Grittnr, K. (2012). Aislamiento de instantáneas serializables en PostgreSQL. *Proc. VLDB. Dotación*, 5(12):1850–1861.
- Pottinger, R. y Levy, AY (2000). Un algoritmo escalable para responder consultas mediante vistas. En *Actas de la 26.ª Conferencia Internacional sobre bases de datos muy grandes*, páginas 484–495.
- Pu, C. (1988). Superbases de datos para la composición de bases de datos heterogéneas. En *Actas de la 4.ª Conferencia Internacional sobre ingeniería de datos*, páginas 548–555.
- Pu, C. y Leff, A. (1991). Control de réplicas en sistemas distribuidos: Un enfoque asíncrono. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*, páginas 377–386.
- Qian, Z., He, Y., Su, C., Wu, Z., Zhu, H., Zhang, T., Zhou, L., Yu, Y. y Zhang, Z. (2013). Timestream: Cálculo fiable de flujos en la nube. En *Actas de la 8.ª Conferencia Europea ACM SIGOPS/EuroSys sobre Sistemas de Computación*, páginas 1–14.
- Qin, L., Yu, J. X., Chang, L., Cheng, H., Zhang, C. y Lin, X. (2014). Procesamiento escalable de grafos grandes en MapReduce. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 827–838.
- Quamar, A., Kumar, KA y Deshpande, A. (2013). Sword: Ubicación escalable de datos con capacidad de respuesta a cargas de trabajo para cargas de trabajo transaccionales. En *Actas de la 16.ª Conferencia Internacional sobre la Extensión de la Tecnología de Bases de Datos*, páginas 430–441.
- Raghavan, S. y García-Molina, H. (2001). Explorando la web oculta. En *Actas de la 27.ª Conferencia Internacional sobre Bases de datos muy grandes*.
- Raghavan, S. y García-Molina, H. (2003). Representación de grafos web. En *Actas de la 19.ª Conferencia Internacional sobre Ingeniería de datos*, páginas 405–416.
- Rahal, A., Zhu, Q. y Larson, P.-A. (2004). Técnicas evolutivas para actualizar el coste de las consultas. *Modelos en un entorno multibase de datos dinámico. VLDB J.*, 13(2):162–176.
- Rahimi, S. (1987). Arquitectura de referencia para sistemas distribuidos de gestión de bases de datos. En *Proc. 3.ª Conferencia Internacional sobre Ingeniería de Datos. Notas del Tutorial*.
- Rahimi, SK y Haug, FS (2010). *Sistemas de gestión de bases de datos distribuidas: una guía práctica*. Enfoque. Wiley.
- Rahm, E. y Bernstein, P. a. (2001). Un estudio de enfoques para la correspondencia automática de esquemas. *VLDB J.*, 10(4):334–350.
- Rahm, E. y Do, HH (2000). Limpieza de datos: Problemas y enfoques actuales. *Q. Bull. IEEE TC sobre ingeniería de datos*, 23(4):3–13.
- Rahm, E. y Marek, R. (1995). Balanceo de carga dinámico multirecurso en bases de datos paralelas. *sistemas. En Actas de la 21.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 395–406.
- Ramabhadran, S., Ratnasamy, S., Hellerstein, JM y Shenker, S. (2004). Breve anuncio: árbol hash de prefijos. En *Proc. ACM SIGACT-SIGOPS 23.º Simposio sobre los Principios de la Computación Distribuida*, página 368.
- Ramamoorthy, CV y Wah, BW (1983). El isomorfismo de la asignación simple de archivos. *IEEE Trans. Comput.*, 32:221–223.
- Ramamritham, K. y Pu, C. (1995). Una caracterización formal de la serialización epsilon. *IEEE Trans. Knowl. and Data Eng.*, 7(6):997–1007.
- Raman, V. y Hellerstein, JM (2001). Torno de alfarero: Un sistema interactivo de limpieza de datos. En *Actas de la 27.ª Conferencia Internacional sobre bases de datos muy grandes*, páginas 381–390.
- Raman, V., Deshpande, A. y Hellerstein, JM (2003). Uso de módulos de estado para consultas adaptativas. *Procesamiento. En Actas de la 19.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 353–365.
- Rao, J., Zhang, C., Megiddo, N. y Lohman, G. (2002). Automatización del diseño de bases de datos físicas en una base de datos paralela. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*.
- Rastogi, V., Machanavajhala, A., Chitnis, L. y Sarma, AD (2013). Detección de componentes conexos en map-reduce en rondas logarítmicas. En *Actas de la 29.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 50–61.
- Ratnasamy, S., Francis, P., Handley, M., Karp, R. y Shenker, S. (2001). Una red escalable y direccionable por contenido. *ACM SIGCOMM Computer Communication Review*, 31(4):161–172.
- Ray, I., Mancini, LV, Jajodia, S. y Bertino, E. (2000). ASEP: Un protocolo de confirmación seguro y flexible para sistemas de bases de datos distribuidas MLS. *IEEE Trans. Knowl. and Data Eng.*, 12(6):880–899.

- Redmond, E. y Wilson, JR (2012). Siete bases de datos en siete semanas: Una guía para las bases de datos modernas Bases de datos y el movimiento NoSQL. Los programadores pragmáticos.
- Reed, DP (1978). Nomenclatura y sincronización en un sistema informático descentralizado. Tesis doctoral.
- Reiss, F. y Hellerstein, J. (2005). Triage de datos: una arquitectura adaptativa para el desastre de carga en TelegraphCQ. En Actas de la 21.ª Conferencia Internacional sobre Ingeniería de Datos, páginas 155-156.
- Rekatsinas, T., Joglekar, M., García-Molina, H., Parameswaran, AG y Ré, C. (2017). SLIMFast: Resultados garantizados para la fusión de datos y la fiabilidad de las fuentes. En Actas de la Conferencia Internacional ACM SIGMOD sobre Gestión de Datos, páginas 1399–1414.
- Revilak, S., O'Neil, PE, y O'Neil, EJ (2011). Aislamiento de instantáneas serializables con precisión (PSSI). En Proc. 27th Int. Conf. sobre Ingeniería de Datos, páginas 482–493.
- Ribeiro-Neto, BA y Barbosa, RA (1998). Rendimiento de consultas para bibliotecas digitales distribuidas y estrechamente acopladas. En Actas de la 3.ª Conferencia Internacional de la ACM sobre Bibliotecas Digitales, páginas 182-190.
- Richter, S., Quiané-Ruiz, J.-A., Schuh, S. y Dittich, J. (2013). Hacia una indexación estática y adaptativa sin sobrecarga en Hadoop. VLDB J., 23(3):469–494.
- Ritter, J. (2001). ¿Por qué Gnutella no puede escalar? No, en serio. <http://www.darkridge.com/~jpr5/doc/gnutella.html> . Último acceso junio de 2019.
- Rivera-Vega, P., Varadarajan, R., y Navathe, SB (1990). Programación de la redistribución de datos en bases de datos distribuidas. En Proc. Int. Conf. on Data Eng., páginas 166-173.
- Rjaibi, W. (2004). Introducción a los sistemas de gestión de bases de datos relacionales seguros multinivel. En Proc. Conf. del Centro IBM de Estudios Avanzados sobre Investigación Colaborativa, páginas 232– 241.
- Robinson, I., Webber, J., y Eifrem, E. (2015). Bases de datos de grafos. O'Reilly, 2.ª edición.
- Röhm, U., Böhm, K., y Schek, H.-J. (2000). Enrutamiento de consultas OLAP y diseño físico en un clúster de bases de datos. En Avances en Tecnología de Bases de Datos, Actas de la 7.ª Conferencia Internacional sobre la Extensión de la Tecnología de Bases de Datos, páginas 254–268.
- Röhm, U., Böhm, K. y Schek, H.-J. (2001). Enrutamiento de consultas con caché en un clúster de bases de datos. En Proc. 17a Conferencia Internacional sobre Ingeniería de Datos, páginas 641–650.
- Röhm, U., Böhm, K., Schek, H.-J. y Schuld, H. (2002). FAS: Un middleware de coordinación sensible a la frescura para un clúster de componentes OLAP. En Actas de la 28.ª Conferencia Internacional sobre Bases de Datos Muy Grandes, páginas 754–765.
- Roitman, H. y Gal, A. (2006). Ontobuilder: Extracción y consolidación totalmente automática de ontologías de fuentes web mediante semántica de secuencias. En Proc. EDBT Workshops, volumen 4254 de LNCS, páginas 573–576.
- Roth, M. y Schwartz, P. (1997). ¡No lo descartes, envuélvelo! Una arquitectura de envoltorio para datos heredados. Fuentes. En Actas de la 23.ª Conferencia Internacional sobre Bases de Datos Muy Grandes, páginas 266–275.
- Roth, MT, Ozcan, F., y Haas, LM (1999). Los modelos de costos son importantes: Proporcionar información de costos para diversas fuentes de datos en un sistema federado. En Actas de la 25.ª Conferencia Internacional sobre Bases de Datos Muy Grandes, páginas 599-610.
- Rothermel, K. y Mohan, C. (1989). Aries/nt: Un método de recuperación basado en el registro de escritura anticipada para transacciones anidadas. En Actas de la 15.ª Conferencia Internacional sobre Bases de Datos Muy Grandes, páginas 337–346.
- Roubini, N. (2018). Testimonio ante la audiencia del Comité de Banca, Vivienda y Asuntos Comunitarios del Senado de EE. UU. sobre la exploración del ecosistema de criptomonedas y blockchain. Accesible en <https://www.banking.senate.gov/imo/media/doc/Roubini%20Testimony%2010-11-18.pdf>/. Último acceso febrero de 2019.
- Roy, A., Mihailovic, I. y Zwaenepoel, W. (2013). X-stream: procesamiento de grafos centrado en el borde mediante particiones de streaming. En Proc. 24.º Simposio ACM sobre Principios de Sistemas Operativos, páginas 472–488.
- Ryvina, E., Maskey, A., Adams, I., Sandler, B., Fuchs, C., Cherniack, M. y Zdonik, S. (2006). Procesamiento de revisiones en un motor de procesamiento de flujos: Un diseño de alto nivel. En Proc. 22.ª edición. Conferencia sobre Ingeniería de Datos, página 141.
- Sacca, D. y Wiederhold, G. (1985). Particionamiento de bases de datos en un clúster de procesadores. ACM Trans. Base de datos Syst., 10(1):29–56.

- Sacco, MS y Yao, SB (1982). Optimización de consultas en sistemas de bases de datos distribuidas. En Yovits, M., editor, *Advances in Computers*, volumen 21, páginas 225–273.
- Saito, Y. y Shapiro, M. (2005). Replicación optimista. *ACM Comput. Surv.*, 37(1):42–81.
- Sakr, S., Liu, A. y Fayoumi, AG (2013). La familia de MapReduce y los sistemas de procesamiento de datos a gran escala. *ACM Comput. Surv.*, 46(1):11:1–11:44.
- Salihoglu, S. y Widom, J. (2013). GPS: un sistema de procesamiento de grafos. En *Actas de la 25.ª Conferencia Internacional sobre Gestión de bases de datos científicas y estadísticas*, páginas 22:1–22:12.
- Salihoglu, S. y Widom, J. (2014). Optimización de algoritmos de grafos en sistemas tipo Pregel. *Proc. Dotación VLDB*, 7(7):577–588.
- Salton, G. (1989). *Procesamiento automático de textos: transformación, análisis y recuperación de información por computadora*. Addison-Wesley.
- Sandhu, RS, Coyne, EJ, Feinstein, HL y Youman, CE (1996). Modelos de control de acceso basados en roles. *IEEE Computer*, 29(2):38–47.
- Schenkel, R., Weikum, G., Weißenberg, N. y Wu, X. (2000). Gestión de transacciones federadas con aislamiento de instantáneas. En Saake, G., Schwarz, K. y Türkler, C., editores, *Transactions and Database Dynamics*, páginas 1–25. Springer.
- Schmachtenberg, M., Bizer, C. y Paulheim, H. (2014). Adopción de buenas prácticas de datos en diferentes ámbitos temáticos. En *Proc. 13.ª Conferencia Internacional sobre Web Semántica*, páginas 245-260.
- Schmidt, C. y Parashar, M. (2004). Facilitación de consultas flexibles con garantías en sistemas P2P. *Computación en Internet del IEEE*, 8(3):19–26.
- Schreiber, F. (1977). Un marco para sistemas de bases de datos distribuidas. En *Proc. Int. Computing Symposium*, páginas 475–482.
- Schuhknecht, FM, Jindal, A., y Dittrich, J. (2013). Las piezas no agrietadas en el agrietamiento de bases de datos. *Proc. Dotación VLDB*, 7(2):97–108.
- Selinger, PG y Adiba, M. (1980). Selección de rutas de acceso en la gestión distribuida de bases de datos. sistemas. En *Proc. Primera Conferencia Internacional sobre Bases de Datos*, páginas 204-215.
- Selinger, PG, Astrahan, MM, Chamberlin, DD, Lorie, RA y Price, TG (1979). Selección de ruta de acceso en un sistema de gestión de bases de datos relacionales. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 23-34.
- Sequeda, JF, Arenas, M., y Miranker, DP (2014). OBDA: ¿reescritura de consultas o materialización? En la práctica, ¡ambos! En *Actas de la 13.ª Conferencia Internacional sobre Web Semántica*, páginas 535-551.
- Shah, MA, Hellerstein, JM, Chandrasekaran, S. y Franklin, MJ (2003). Flux: Un operador de partición adaptativo para sistemas de consulta continua. En *Actas de la 19.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 25-36.
- Shao, B., Wang, H. y Li, Y. (2013). Trinity: un motor de gráficos distribuido en una nube de memoria. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*, páginas 505–516.
- Shatdal, A. y Naughton, JF (1993). Uso de memoria virtual compartida para el procesamiento de uniones paralelas. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*, páginas 119–128.
- Shatdal, A., Kant, C. y Naughton, J.F. (1994). Algoritmos con caché para el procesamiento de consultas relacionales. En *Proc. 20.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 510–521.
- Shekita, EJ, Young, HC y Tan, KL (1993). Optimización multiunión para multiprocesadores simétricos. En *Actas de la 19.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 479–492.
- Sheth, A., Larson, J., Cornello, A. y Navathe, SB (1988a). Una herramienta para integrar esquemas conceptuales y vistas de usuario. En *Actas de la 4.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 176-183.
- Sheth, A., Larson, J. y Watkins, E. (1988b). Tailor, una herramienta para actualizar vistas. En *Avances en Tecnología de Bases de Datos*, *Actas de la 1.ª Conferencia Internacional sobre la Extensión de la Tecnología de Bases de Datos*, páginas 190-213.
- Sheth, AP y Kashyap, V. (1992). Tan lejos (esquemáticamente) pero tan cerca (semánticamente). En *Proc. Conferencia sobre semántica de bases de datos del IFIP WG 2.6 sobre sistemas de bases de datos interoperables*, páginas 283–312.
- Sheth, AP y Larson, J. (1990). Sistemas de bases de datos federadas para la gestión de bases de datos distribuidas, heterogéneas y autónomas. *ACM Comput. Surv.*, 22(3):183–236.
- Shute, J., Vingralek, R., Samwel, B., Handy, B., Whipkey, C., Rollins, E., Oancea, M., Littlefield, K., Menestrina, D., Ellner, S., Cieslewicz, J., Rae, I., Stancescu, T. y Apte, H. (2013). F1: Una base de datos SQL distribuida escalable. *Proc. VLDB Endowment*, 6(11):1068–1079.

- Sidell, J., Aoki, PM, Sah, A., Staelin, C., Stonebraker, M. y Yu, A. (1996). Replicación de datos en Mariposa. En *Actas de la 12.ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 485–494.
- Sidirourgos, L., Goncalves, R., Kersten, M., Nes, N. y Manegold, S. (2008). Soporte de almacenamiento en columnas para la gestión de datos RDF: no todos los cisnes son blancos. *Proc. VLDB Endowment*, 1(2): 1553–1563.
- Silberschatz, A., Korth, H. y Sudarshan, S. (2019). *Conceptos de sistemas de bases de datos*. McGraw-Hill, 7ma edición.
- Simitsis, A., Wilkinson, K., Castellanos, M. y Dayal, U. (2009). Diseño ETL basado en QoX: reducción del coste de las consultorías ETL. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 953–960.
- Simitsis, A., Wilkinson, K., Castellanos, M. y Dayal, U. (2012). Optimización de flujos de datos analíticos para múltiples motores de ejecución. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 829–840.
- Simon, E. y Valduriez, P. (1984). Diseño e implementación de un subsistema de integridad extensible. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 9–17.
- Simon, E. y Valduriez, P. (1986). Control de integridad en sistemas de bases de datos distribuidas. En *Proc. 19.ª edición. Conferencia Internacional de Hawái sobre Ciencias de Sistemas*, páginas 622–632.
- Simon, E. y Valduriez, P. (1987). Diseño y análisis de un subsistema de integridad relacional. Informe técnico DB-015-87, Microelectronics and Computer Corporation, Austin, Texas.
- Singhal, M. (1989). Detección de interbloqueos en sistemas distribuidos. *Computer*, 22(11):37–48.
- Skarra, A. (1989). Control de concurrencia para transacciones cooperantes en una base de datos orientada a objetos. En *Proc. Taller ACM SIGPLAN sobre programación concurrente basada en objetos*, páginas 145– 147.
- Skarra, A., Zdonik, S. y Reiss, S. (1986). Un servidor de objetos para un sistema de bases de datos orientado a objetos. En *Actas del 1.er Taller Internacional sobre Sistemas de Bases de Datos Orientados a Objetos*, páginas 196–204.
- Skeen, D. (1981). Protocolos de confirmación no bloqueantes. En la *Conferencia Internacional sobre Gestión de ACM SIGMOD de Datos*, páginas 133–142.
- Skeen, D. (1982a). Un protocolo de confirmación basado en quórum. En *Actas del 6.º Taller de Berkeley sobre Gestión Distribuida de Datos y Redes Informáticas*, páginas 69–80.
- Skeen, D. (1982b). Recuperación ante fallos en un sistema de gestión de bases de datos distribuidas. Tesis doctoral, Departamento de Ingeniería Eléctrica y Ciencias de la Computación, Universidad de California en Berkeley, Berkeley, California.
- Skeen, D. y Stonebraker, M. (1983). Un modelo formal de recuperación ante fallos en un sistema distribuido. Traducción IEEE. *Software. Ing.*, SE-9(3):219–228.
- Skeen, D. y Wright, D. (1984). Aumento de la disponibilidad en redes particionadas. En *Actas del 3.er Simposio ACM SIGACT-SIGMOD sobre Principios de Sistemas de Bases de Datos*, páginas 290–299.
- Somani, A., Choy, D. y Kleewein, JC (2002). Integración de sistemas de gestión de contenido y datos: Desafíos y oportunidades. *IBM Systems J.*, 41(4):686–696.
- Sousa, A., Oliveira, R., Moura, F. y Pedone, F. (2001). Replicación parcial en la máquina de estados de la base de datos. En *Proc. IEEE Int. Symp. Network Computing and Applications*, páginas 298–309.
- Srivastava, U. y Widom, J. (2004a). Gestión flexible del tiempo en sistemas de flujo de datos. En *Proc. Simposio ACM SIGACT-SIGMOD sobre principios de sistemas de bases de datos*, páginas 263–274.
- Srivastava, U. y Widom, J. (2004b). Ejecución con memoria limitada de uniones de flujo con ventana. En *Actas de la 30.ª Conferencia Internacional sobre bases de datos muy grandes*, páginas 324–335.
- Stanoi, I., Agrawal, D., y El Abbadi, A. (1998). Uso de primitivas de difusión en bases de datos replicadas. En *Actas de la 8.ª Conferencia Internacional del IEEE sobre Sistemas de Computación Distribuida*, páginas 148–155.
- Stöhr, T., Mörtens, H. y Rahm, E. (2000). Asignación de bases de datos multidimensionales para almacenes de datos paralelos. En *Actas de la 26.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 273–284.
- Stoica, I., Morris, R., Karger, D., Kaashoek, M. y Balakrishnan, H. (2001). Chord: Un servicio de búsqueda escalable entre pares para aplicaciones de internet. En *Proc. 2001, Conferencia sobre Aplicaciones, Tecnologías, Arquitecturas y Protocolos para la Comunicación Informática*, páginas 149–160.



- Stoica, I., Morris, R., Liben-Nowell, D., Karger, D., Kaashoek, M., Dabek, F. y Balakrishnan, H. (2003). Chord: un protocolo escalable de búsqueda entre pares para aplicaciones de internet. *IEEE/ACM Trans. Netw.*, 11(1):17–32.
- Stonebraker, M. (1975). Implementación de restricciones de integridad y vistas mediante modificación de consultas. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*, páginas 65–78.
- Stonebraker, M. (1981). Soporte de sistemas operativos para la gestión de bases de datos. *Commun. ACM*, 24(7):412–418.
- Stonebraker, M. (1986). El argumento a favor de la nada compartida. *Q. Bull. IEEE TC on Data Eng.*, 9(1):4–9.
- Stonebraker, M. y Neuhold, E. (1977). Una versión de base de datos distribuida de INGRES. En *Actas del 2.º Taller de Berkeley sobre Gestión de Datos Distribuidos y Redes de Computadoras*, páginas 9–36.
- Stonebraker, M., Abadi, DJ, DeWitt, DJ, Madden, S., Paulson, E., Pavlo, A. y Rasin, A. (2010). MapReduce y sistemas de gestión de bases de datos paralelos: ¿amigos o enemigos? *Commun. ACM*, 53(1):64–71.
- Strauch, C. (2011). Bases de datos NoSQL. Universidad de Medios de Comunicación de Stuttgart.
- Sullivan, M. y Heybey, A. (1998). Tribeca: Un sistema para la gestión de grandes bases de datos de red. tráfico. En *Proc. Conferencia Técnica Anual USENIX 1998*
- Swami, A. (1989). Optimización de consultas de unión de gran tamaño: combinación de heurísticas y técnicas combinatorias. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 367–376.
- Taft, R., Mansour, E., Serafini, M., Duggan, J., Elmore, A.J., Aboulmaga, A., Pavlo, A. y Stonebraker, M. (2014). E-Store: Particionamiento elástico de grano fino para el procesamiento de transacciones distribuidas. *Proc. VLDB Endowment*, 8(3):245–256.
- Taft, R., El-Sayed, N., Serafini, M., Lu, Y., Aboulmaga, A., Stonebraker, M., Mayerhofer, R. y Andrade, F. (2018). P-store: Un sistema de base de datos elástica con aprovisionamiento predictivo. En *Proc. Conferencia internacional ACM SIGMOD sobre gestión de datos*, páginas 205–219.
- Tandem. (1987). NonStop SQL: una implementación distribuida de SQL de alto rendimiento y alta disponibilidad. En *Proc. Int. Workshop on High Performance Transaction Systems*, páginas 60–104.
- Tandem. (1988). Un punto de referencia de NonStop SQL en la transacción débito-crédito. En *Proc. ACM SIGMOD Int. Conf. on Management of Data*, páginas 337–341.
- Tanenbaum, AS y van Renesse, R. (1988). Votación con fantasmas. En *Actas de la 8.ª Conferencia Internacional del IEEE sobre sistemas de computación distribuida*, páginas 456–461.
- Tang, W., Zhao, X., Rafique, W., Qi, L., Dou, W. y Ni, Q. (2019). Un método de descarga que utiliza servidores móviles de borde descentralizados con tecnología P2P en la computación de borde. *Journal of Systems Architecture – Embedded Systems Design*, 94:1–13.
- Tatarinov, I., Ives, ZG, Madhavan, J., Halevy, AY, Suciu, D., Dalvi, NN, Dong, X., Kadiyska, Y., Miklau, G. y Mork, P. (2003). El proyecto de gestión de datos entre pares de Piazza. *Rec. ACM SIGMOD*, 32(3):47–52.
- Tatbul, N., Cetintemel, U., Zdonik, S., Cherniack, M. y Stonebraker, M. (2003). Deslastre de carga en un gestor de flujo de datos. En *Actas de la 29.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 309–320.
- Thiran, P., Hainaut, J.-L., Houben, G.-J. y Benslimane, D. (2006). Evolución de sistemas de información heredados basada en wrappers. *ACM Trans. Softw. Eng. and Meth.*, 15(4):329–359.
- Thomas, RH (1979). Un enfoque de consenso mayoritario para el control de concurrencia en copias múltiples. bases de datos. *ACM Trans. Database Syst.*, 4(2):180–209.
- Thomasian, A. (1996). Control de concurrencia de bases de datos: métodos, rendimiento y análisis. Editores académicos Kluwer.
- Thomson, A. y Abadi, DJ (2010). El caso del determinismo en sistemas de bases de datos. *Proc. VLDB Dotación*, 3(1):70–80.
- Thuraisingham, B. (2001). Sistemas de bases de datos distribuidos seguros. Seguridad de la información. Informe, 6(2).
- Thursoo, A., Sarma, JS, Jain, N., Shao, Z., Chakka, P., Anthony, S., Liu, H., Wyckoff, P. y Murthy, R. (2009). Hive: una solución de almacenamiento sobre un marco de map-reduce. *Proc. VLDB Endowment*, 2(2):1626–1629.
- Tian, F. y DeWitt, DJ (2003). Estrategias de enrutamiento de tuplas para remolinos distribuidos. En *Proc. 29.ª edición. Conferencia sobre bases de datos muy grandes*, páginas 333–344.
- Tian, Y., Balmin, A., Corsten, SA, Tatikonda, S. y McPherson, J. (2013). De “piensa como un vértice” a “pensar como un grafo”. *Proc. VLDB Endowment*, 7(3):193–204.

- Tian, Y., Özcan, F., Zou, T., Goncalves, R. y Piraheh, H. (2016). Construcción de un almacén híbrido: Uniones eficientes entre datos almacenados en HDFS y un almacén empresarial. *ACM Trans. Database Syst.*, 41(4):21:1–21:38.
- Tomasic, A., Raschid, L. y Valduriez, P. (1996). Escalado de bases de datos heterogéneas y diseño de discos. En *Actas de la 16.ª Conferencia Internacional del IEEE sobre Sistemas de Computación Distribuida*, páginas 449–457.
- Tomasic, A., Amouroux, R., Bonnet, P., Kapitskaia, O., Naacke, H. y Raschid, L. (1997). El componente de búsqueda de información distribuida (DISCO) y la World Wide Web: demostración de prototipo. En *Actas de la Conferencia Internacional sobre Gestión de Datos de ACM SIGMOD*, páginas 546–548.
- Tomasic, A., Raschid, L. y Valduriez, P. (1998). Escalado del acceso a sistemas heterogéneos distribuidos. *Fuentes de datos con Disco*. En *IEEE Trans. Knowl. y Data Eng.* en prensa.
- Toshniwal, A., Taneja, S., Shukla, A., Ramasamy, K., Patel, JM, Kulkarni, S., Jackson, J., Gade, K., Fu, M., Donham, J., Bhagat, N., Mittal, S. y Ryaboy, D. (2014). Tormenta@twitter. En *Proc. Conferencia internacional ACM SIGMOD sobre gestión de datos*, páginas 147–156.
- Traiger, IL, Gray, J., Galtieri, CA, y Lindsay, BG (1982). Transacciones y recuperación en sistemas de bases de datos distribuidas. *ACM Trans. Database Syst.*, 7(3):323–342.
- Triantafillou, P. y Taylor, DJ (1995). El paradigma basado en la localización para la replicación: Logrando eficiencia y disponibilidad en sistemas distribuidos. *IEEE Trans. Softw. Eng.*, 21(1):1–18.
- Tu, S., Zheng, W., Kohler, E., Liskov, B. y Madden, S. (2013). Transacciones rápidas en bases de datos multinúcleo en memoria. En *Proc. 24.º Simposio ACM sobre Principios de Sistemas Operativos*, páginas 18–32.
- Tucker, P., Maier, D., Sheard, T. y Faragas, L. (2003). Explotación de la semántica de puntuación en flujos de datos continuos. *IEEE Trans. Knowl. and Data Eng.*, 15(3):555–568.
- Ugander, J. y Backstrom, L. (2013). Propagación de etiquetas balanceada para particionar grafos masivos. En *Proc. 6th ACM Int. Conf. Búsqueda web y minería de datos*, páginas 507–516.
- Ullman, J. (1997). Integración de información mediante vistas lógicas. En *Actas de la 6.ª Conferencia Internacional sobre Teoría de Bases de Datos*, págs. 19–40.
- Ullman, JD (1982). Principios de sistemas de bases de datos. Computer Science Press, 2.ª edición.
- Ulusoy, Ö. (2007). Cuestiones de investigación en la gestión de datos entre pares. En *Actas del 22.º Simposio Internacional sobre Ciencias de la Computación y la Información*, páginas 1–8.
- Umbrich, J., Hose, K., Karnstedt, M., Harth, A. y Polleres, A. (2011). Comparación de resúmenes de datos para el procesamiento de consultas en tiempo real sobre datos enlazados. *World Wide Web J.*, 14(5-6):495–544.
- Urhan, T. y Franklin, M. (2000). XJoin: Un operador de unión en pipeline programado reactivamente. *Q. Bull. IEEE TC sobre ingeniería de datos*, 23:27.
- Urhan, T., Franklin, MJ, y Amsaleg, L. (1998). Codificación de consultas basada en costos para retrasos iniciales. En *Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos*, páginas 130–141.
- Valduriez, P. (1982). Algoritmos de semiunión para máquinas de bases de datos distribuidas. En Schneider, J.-J., editor, *Distributed Data Bases*, páginas 23–37.
- Valduriez, P. (1993). Sistemas de bases de datos paralelas: Problemas abiertos y nuevas cuestiones. *Distribuido en paralelo. Bases de datos*, 1:137–16.
- Valduriez, P. y Gardarin, G. (1984). Algoritmos de unión y semiunión para una base de datos multiprocesador. *máquina. ACM Trans. Database Syst.*, 9(1):133–161.
- Valduriez, P. y Pacitti, E. (2004). Gestión de datos en sistemas P2P a gran escala. En *Proc. 6.ª edición. Conf. Computación de alto rendimiento para ciencias computacionales*, páginas 104–118.
- Valiant, LG (1990). Un modelo puente para computación paralela. *Commun. ACM*, 33(8):103–111. van Hee, K. (2002). Gestión del flujo de trabajo. MIT Press.
- Van Renesse, R. y Altinbuken, D. (2015). Paxos moderadamente complejo. *ACM Comput. Surv.*, 47(3):42:1–42:36.
- Varadarajan, R., Rivera-Vega, P., y Navathe, SB (1989). Programación de la redistribución de datos en redes totalmente conectadas. En *Actas de la 27.ª Conferencia Anual Allerton sobre Comunicación, Control y Computación*.
- Velegarakis, Y., Miller, RJ, y Popa, L. (2004). Preservación de la consistencia del mapeo bajo esquema cambios. *Vldb J.*, 13(3):274–293.
- Verhofstadt, JS (1978). Técnicas de recuperación para sistemas de bases de datos. *ACM Comput. Surv.*, 10(2): 168–195.

- Verma, S., Leslie, LM, Shin, Y. y Gupta, I. (2017). Una comparación experimental de estrategias de partición en el procesamiento distribuido de grafos. *Proc. VLDB Endowment*, 10(5):493–504.
- Vermeer, M. (1997). Interoperabilidad semántica para bases de datos heredadas. Tesis doctoral, Departamento de Ciencias de la Computación, Universidad de Twente, Enschede, Países Bajos.
- Viglas, S., Naughton, J. y Burger, J. (2003). Maximización de la tasa de salida de consultas multi-join sobre fuentes de información en streaming. En *Actas de la 29.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 285–296.
- Voulgaris, S., Jelasity, M. y van Steen, M. (2003). Un protocolo robusto y escalable de intercambio de información entre pares. En *Agentes y Computación entre pares, Segundo Taller Internacional (AP2PC)*, páginas 47–58.
- Vu, QH, Lupu, M. y Ooi, BC (2009). Computación peer-to-peer: principios y aplicaciones. Saltador.
- Wah, BW y Lien, YN (1985). Diseño de bases de datos distribuidas en sistemas informáticos locales. Traducción IEEE. *Software. Ing.*, SE-11(7):609–619.
- Walton, C., Dale, A. y Jenevin, R. (1991). Taxonomía y modelo de rendimiento de los efectos de sesgo de datos en uniones paralelas. En *Actas de la 17.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 537–548.
- Wang, G., Xie, W., Demers, AJ y Gehrke, J. (2013). Procesamiento asíncrono de grafos a gran escala simplificado. En *Actas de la 6.ª Conferencia Bienal sobre Investigación Innovadora en Sistemas de Datos*.
- Wang, H., Zaniolo, C. y Luo, R. (2003). Atlas: Una extensión SQL pequeña pero completa para minería de datos y flujos de datos. En *Actas de la 29.ª Conferencia Internacional sobre Bases de Datos Muy Grandes*, páginas 1113–1116.
- Wang, L., Xiao, Y., Shao, B. y Wang, H. (2014). Cómo particionar un grafo de mil millones de nodos. En *Proc. 30ª Conferencia Internacional sobre Ingeniería de Datos*, páginas 568–579.
- Wang, W., Li, J., Zhang, D. y Guo, L. (2004). Procesamiento de agregados de unión de ventana deslizante en consultas continuas sobre flujos de datos. En *Proc. 8.ª Conferencia de Europa del Este. Avances en Bases de Datos y Sistemas de Información*, páginas 348–363.
- Weikum, G. y Vossen, G. (2001). *Sistemas de información transaccional: teoría, algoritmos y La práctica del control de concurrencia*. Morgan Kaufmann.
- Weil, SA, Brandt, SA, Miller, EL, Long, DDE y Maltzahn, C. (2006). Ceph: Un sistema de archivos distribuido escalable y de alto rendimiento. En *Actas del 7.º Simposio USENIX sobre Diseño e Implementación de Sistemas Operativos*, páginas 307–320.
- Weiss, C., Karras, P. y Bernstein, A. (2008). Hexastore: indexación séxtuple para la gestión de datos de la web semántica. *Proc. VLDB Endowment*, 1(1):1008–1019.
- Wiederhold, G. (1992). Mediadores en la arquitectura de los sistemas de información del futuro. *Computer*, 25(3):38–49.
- Wiesmann, M., Schiper, A., Pedone, F., Kemme, B. y Alonso, G. (2000). Técnicas de replicación de bases de datos: Una clasificación de tres parámetros. En *Proc. 28.º Simposio sobre Sistemas Distribuidos Confiables*, páginas 206–215.
- Wilkinson, K. (2006). Implementación de la tabla de propiedades de Jena. Informe técnico HPL-2006-140, HP Laboratorios Palo Alto.
- Wilms, PF y Lindsay, BG (1981). Un mecanismo de autorización de bases de datos que facilita la autorización individual y grupal. Informe de investigación RJ 3137, Laboratorio de Investigación IBM Almaden, San José, California.
- Wilschut, A. y Apers, P. (1991). Ejecución de consultas de flujo de datos en un entorno paralelo de memoria principal. En *Proc. 1.ª Conferencia Internacional sobre Sistemas de Información Paralelos y Distribuidos*, páginas 68–77.
- Wilson, B. y Navathe, SB (1986). Un marco analítico para el rediseño de bases de datos distribuidas. En *Actas del 6.º Simposio de Bases de Datos Avanzadas*, págs. 77–83.
- Wolfson, O. (1987). La sobrecarga de los protocolos de bloqueo (y confirmación) en bases de datos distribuidas. *ACM Trans. Base de datos Syst.*, 12(3):453–471.
- Wong, E. (1977). Recuperación de datos dispersos de SDD-1. En *Actas del 2.º Taller de Berkeley sobre Gestión distribuida de datos y redes de computadoras*, páginas 217–235.
- Wong, E. y Youssefi, K. (1976). Descomposición: Una estrategia para el procesamiento de consultas. *ACM Trans. Base de datos Syst.*, 1(3):223–241.

- Wright, DD (1983). Gestión de bases de datos distribuidas en redes particionadas. Informe técnico. TR83-572, Departamento de Ciencias de la Computación, Universidad de Cornell, Ithaca, NY
- Wu, E., Diao, Y. y Rizvi, S. (2006). Procesamiento de eventos complejos de alto rendimiento sobre flujos. En Proc. ACM SIGMOD Int. Conf. sobre Gestión de Datos, páginas 407–418.
- Wu, K.-L., Yu, PS, y Pu, C. (1997). Algoritmos de control de divergencia para serialización  $\epsilon$ . IEEE Trans. Conocimiento e Ingeniería de Datos, 9(2):262–274.
- Wu, S., Yu, G., Yu, Y., Ou, Z., Yang, X. y Gu, Y. (2005). Un enfoque sensible a plazos para el procesamiento en tiempo real de ventanas deslizantes. En Actas de la 6.ª Conferencia Internacional sobre Gestión de la Información en la Era Web, páginas 566–577.
- Xing, Y., Hwang, J.-H., Çetintemel, U. y Zdonik, S. (2006). Proporcionando resiliencia a las variaciones de carga en el procesamiento de flujos distribuidos. En Actas de la 32.ª Conferencia Internacional sobre Bases de Datos Muy Grandes, páginas 775–786.
- Yan, D., Cheng, J., Lu, Y. y Ng, W. (2014). Blogel: Un marco centrado en bloques para computación distribuida en grafos del mundo real. Proc. VLDB Endowment, 7(14):1981–1992.
- Yan, D., Bu, Y., Tian, Y. y Deshpande, A. (2017). Plataformas de análisis de grafos grandes. Fundamentos y tendencias en bases de datos, 7(1-2):1–195.
- Yan, LL (1997). Hacia una mediación eficiente y escalable: El enfoque AURORA. En Proc. Conferencia IBM CASCON, páginas 15–29.
- Yan, L.-L., Özsu, MT, y Liu, L. (1997). Acceso a datos heterogéneos mediante mediadores de homogeneización e integración. En Proc. Int. Conf. on Cooperative Inf. Syst., páginas 130-139.
- Yan, LL, Miller, RJ, Haas, LM y Fagin, R. (2001). Comprensión y refinamiento de mapeos de esquemas basados en datos. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 485–496.
- Yang, B. y García-Molina, H. (2002). Mejora de la búsqueda en redes peer-to-peer. En Proc. 22.ª edición. IEEE Int. Conf. sobre sistemas de computación distribuida, páginas 5–14.
- Yang, X., Lee, M.-L. y Ling, T.W. (2003). Resolución de conflictos estructurales en la integración de esquemas XML: Un enfoque semántico. En Actas de la 22.ª Conferencia Internacional sobre Modelado Conceptual, páginas 520–533.
- Yao, SB, Waddle, V. y Housel, B. (1982). Modelado e integración de vistas mediante el modelo funcional. modelo de datos. Traducción IEEE. Software. Ing., SE-8(6):544–554.
- Yu, C. y Meng, W. (1998). Principios de procesamiento de consultas para aplicaciones avanzadas de bases de datos. Morgan Kaufmann.
- Zaharia, M. (2016). Una arquitectura para el procesamiento rápido y general de datos en grandes clústeres. Libros ACM.
- Zaharia, M., Chowdhury, M., Franklin, MJ, Shenker, S. y Stoica, I. (2010). Spark: Computación en clúster con conjuntos de trabajo. En Actas del 2.º Taller USENIX sobre temas de actualidad en computación en la nube, páginas 10-10.
- Zaharia, M., Das, T., Timothy Hunter, HL, Shenker, S. y Stoica, I. (2013). Flujos discretizados: Computación de flujos con tolerancia a fallos a escala. En Proc. 24.º Simposio ACM sobre Principios de Sistemas Operativos, páginas 423–438.
- Zhao, B., Huang, L., Stribling, J., Rhea, S., Joseph, AD, y Kubiawicz, J. (2004). Tapestry: Una superposición resiliente a escala global para la implementación de servicios. IEEE J. Selected Areas in Comm., 22 (1):41–53.
- Zhu, M. y Risch, T. (2011). Consultas a bases de datos relacionales y en la nube. En Proc. Conferencia Internacional sobre Competencia en la Nube y Servicios de 2011, páginas 330–335.
- Zhu, Q. (1995). Estimación de parámetros de coste local para la optimización de consultas globales en un sistema multibase de datos. Tesis doctoral, Departamento de Ciencias de la Computación, Universidad de Waterloo, Waterloo, Canadá.
- Zhu, Q. y Larson, P.-Å. (1994). Un método de muestreo de consultas para la estimación de parámetros de costes locales en un sistema multibase de datos. En Actas de la 10.ª Conferencia Internacional sobre Ingeniería de Datos, páginas 144-153.
- Zhu, Q. y Larson, PA (1996a). Procesamiento y optimización de consultas globales en el sistema multibase de datos CORDS. En Proc. Int. Conf. on Parallel and Distributed Computing Systems, páginas 640–647.

- Zhu, Q. y Larson, PA (1996b). Desarrollo de modelos de regresión de costes para sistemas multibase de datos. En Proc. 4ta Conf. Int. sobre Sistemas de Información Paralelos y Distribuidos, páginas 220–231.
- Zhu, Q. y Larson, PA (1998). Solución del problema de estimación de costes locales para la optimización de consultas globales en sistemas multibase de datos. *Distributed Parall. Databases*, 6(4):373–420.
- Zhu, Q., Sun, Y. y Motheramgari, S. (2000). Desarrollo de modelos de costes con variables cualitativas para entornos dinámicos multibase de datos. En Actas de la 16.ª Conferencia Internacional sobre Ingeniería de Datos, páginas 413–424.
- Zhu, Q., Motheramgari, S., y Sun, Y. (2003). Estimación de costes para consultas con múltiples estados de contención en entornos dinámicos multibase de datos. *Knowledge and Information Systems*, 5(1):26–49.
- Zhu, Y., Rundensteiner, E. y Heineman, G. (2004). Migración dinámica de planes para consultas continuas sobre flujos de datos. En Proc. ACM SIGMOD Int. Conf. on Management of Data, páginas 431–442.
- Zhu, Y., Zhang, H., Qin, L. y Cheng, H. (2017). Algoritmos eficientes de MapReduce para triángulos. Listado en grafos a escala de mil millones. *Distribuidor Paralelo. Bases de Datos*, 35(2):149–176.
- Ziane, M., Zait, M., y Borla-Salamet, P. (1993). Procesamiento de consultas paralelas con árboles en zigzag. *VLDB J.*, 2(3):277–301.
- Zilio, DC (1998). Algoritmos de decisión para el diseño de bases de datos físicas y reorganización concurrente para sistemas de bases de datos paralelas. Tesis doctoral, Universidad de Toronto.
- Zou, L. y Özsu, MT (2017). Gestión de datos RDF basada en grafos. *Ciencia de datos y Ingeniería*, 2(1):56–70.
- Zou, L., Mo, J., Chen, L., Özsu, MT y Zhao, D. (2011). gStore: respuesta a consultas SPARQL mediante correspondencia de subgrafos. *Proc. VLDB Endowment*, 4(8):482–493.
- Zou, L., Özsu, MT, Chen, L., Shen, X., Huang, R. y Zhao, D. (2014). gStore: Un almacén basado en gráficos Motor de consultas SPARQL. *VLDB J.*, 23(4):565–590.

# Índice

Simbolos	Sitio de aprendizaje, 165
2PC, véase Confirmación en dos fases	ArangoDB, 536
3PC, ver Compromiso trifásico	ARTEMISA, 335
	AstérixDB, 528, 556
A	Paralelo asíncrono, 496
Abortar, 185	Semántica de al menos una vez, 485
Control de acceso, 91, 92, 102	Semántica de "como máximo una vez", 485
Frecuencia de acceso, 40	Compromiso atómico, 211
Ruta de acceso, 24	Matriz de afinidad de atributos, 55, 56, 60
Selector de ruta de acceso, 24	Medida de afinidad de atributos, 53, 54
Propiedades del ácido, 185	Valor de uso del atributo, 53
Transacción ACID, 532, 533, 536, 538	Aurora, 471, 479
Cola de activación, 381	Sistema de integración de datos AURORA, 285
AdaptCache, 80	Aurora DSMS, 471, 474, 513, 514
Procesamiento adaptativo de consultas, 174	Autorización, 91
Reacción adaptativa, 175	Matriz de autorización, 104
Particionamiento virtual adaptativo, 389	Detección automática, 611
Entrega de datos ad hoc, 6	Autonomía, 18
Afijo, 292	comunicación, 309
Afirmación agregada, 121	diseño, 309
Restricción agregada, 117	ejecución, 309
Asignación, 33, 43, 49, 51, 67–69, 72, 83, 85, 89	Autoplex, 295
Espectro de desplazamiento al rojo de Amazon, 556	Disponibilidad, 15
Amazon SimpleDB, 524	AVP, consulte Particionamiento virtual adaptativo
Ambari, 511	jIMPRESIONANTE, 556
Consulta de gráficos analíticos, 487	Azure HDInsight, 454
Carga de trabajo de gráficos analíticos, 487	
AP, ver Paralelo asíncrono	B
Apache Flink, 485	Retroenlace, 563
Jirafa apache, 498, 515	Ancho de banda, 11
Apache Ignite, 537	BAP, ver Paralelo asíncrono sin barreras
Tormenta Apache, 471, 482, 514	Paralelo asíncrono sin barreras, 502, 503
APPA, 410, 411, 419, 421, 432, 434, 435, 446	Relación de base, 92
Servidor de aplicaciones, 21	Paxos básico, ver Paxos

- BATON, [406](#), [426](#), [428](#)
- BATON\*, [406](#)
- Conflicto de comportamiento, [289](#)
- Restricción de comportamiento, [110](#)
- Número de Bell, [53](#)
- Algoritmo de mejor posición, [417](#)
- BigchainDB, [444](#)
- Big, [data](#), [3](#), [16](#), [20](#), [449](#)
- aplicación, [449](#)
- procesamiento, [449](#)
- sistema de procesamiento, [449](#)
- BigDAWG, [549](#), [553](#)–[556](#)
- BigIntegrator, [541](#), [542](#), [554](#), [556](#)
- Bigtable, véase Google Bigtable
- Tabla binaria, [604](#), [617](#)
- Bio2RDF, [595](#)
- Bitcoin, [436](#), [439](#)
- Bitcoin-NG, [443](#)
- BitTorrent, [396](#), [399](#), [444](#)
- Almacenamiento basado en bloques, [452](#)
- BLOCKBENCH, [444](#)
- Modelo de gráfico centrado en bloques, [495](#)
- Blockchain, [436](#), [444](#)
- sin permiso, [438](#)
- con permiso, [438](#)
- privado, [438](#)
- público, [438](#)
- Blockchain 2.0, [442](#)
- Blogel, [505](#)
- Algoritmo de energía de enlace, [56](#)
- Borealis, [471](#), [514](#)
- Diseño de abajo hacia arriba, [16](#), [281](#), [283](#)
- BPA, véase Algoritmo de mejor posición
- BSP, véase Índice de árbol B
- paralelo síncrono masivo, [360](#), [363](#)
- Algoritmo de cubo, [315](#)
- Paralelo síncrono masivo, [495](#), [496](#), [502](#)
- Trie de unión bushy, [150](#)
- Trie de consulta bushy, [371](#)
- do
- Administrador de caché, [24](#)
- Consulta de cálculo, [129](#)
- CAN, [404](#), [429](#)
- Cubierta del conjunto de candidatos, [301](#)
- Modelo de datos canónicos, [284](#)
- Teorema CAP, [520](#), [521](#), [554](#)–[556](#)
- Cassandra, [524](#), [529](#), [531](#)
- Lenguaje de consulta de Cassandra, [529](#)
- Catálogo, [13](#), [82](#)
- Catalizador, [551](#)
- Agrupamiento causal, [534](#)
- Consistencia causal, [534](#)
- Cefalo, [454](#)
- Particionado encadenado, [361](#)
- Consulta encadenada, [156](#)
- Camaleón-db, [605](#), [617](#)
- Acorde, [405](#), [523](#)
- Trozo, [453](#)
- ALMEJAS, [617](#)
- Operador de limpieza, [307](#)
- Gerente de clientes, [353](#)
- Cliente/servidor, [4](#), [17](#), [19](#), [21](#), [23](#), [24](#)
- varios clientes/varios servidores, [21](#)
- clientes/un solo servidor, [21](#)
- Nube, [17](#), [455](#)
- Computación en la nube, [27](#), [451](#)
- CloudMdsQL, [539](#), [549](#), [551](#), [552](#), [554](#)–[556](#)
- Matriz de afinidad agrupada, [56](#), [57](#), [60](#), [61](#), [63](#), [64](#), [87](#)
- Agrupamiento, [56](#)
- CucarachaDB, [537](#)
- Almacén de columnas, [36](#)
- COMA, [292](#)
- Compromiso, [185](#)
- Estado comprometido, [226](#)
- Protocolo de confirmación, [211](#)
- Costo de comunicación, [134](#)
- Tiempo de comunicación, [158](#)
- Complejidad de los operadores del álgebra relacional, [134](#)
- Coincidencia compuesta, [295](#)
- Control de concurrencia, [185](#), [239](#)
- bloqueado, [189](#)
- optimista, [189](#)
- pesimista, [189](#)
- ordenación de marcas de tiempo, [189](#)
- Entrega de datos condicional, [6](#)
- Consulta conjuntiva, [312](#)
- Gráfico de conexión, [137](#)
- Consistencia fuerte, [248](#)
- débil, [248](#)
- Hashing consistente, [523](#)
- Coincidencia basada en restricciones, [293](#)
- Borde de contención, [294](#)
- Modelo de procesamiento continuo, [471](#)
- Consulta continua, [6](#)
- Lenguaje de consulta continua, [474](#), [513](#)
- Tiempo muerto del coordinador, [220](#)
- Cosmos DB, [536](#)
- Funciones de costos, [157](#)
- Modelo de costos, [132](#)–[135](#), [157](#), [169](#), [178](#), [269](#), [319](#)–[322](#), [324](#), [329](#), [336](#), [369](#), [372](#), [373](#), [376](#)–[379](#), [391](#), [393](#), [551](#)–[554](#)
- distribuido, [157](#)
- heterogéneo, [317](#), [324](#), [325](#), [336](#)
- mediador, [319](#), [346](#)

Base de sofá, [528](#), [556](#)  
CouchDB, [528](#)  
PUMA, [473](#), [514](#)  
Algoritmo de conteo, [98](#)  
Costo de CPU, [134](#)  
CQL, véase lenguaje de consulta Cassandra  
Crawler, 562–564  
    enfocado, [565](#)  
    incremental, [565](#)  
    paralelo, [566](#)  
Gateando, [585](#)  
Criptomoneda, [436](#)  
Consulta Cúcnica, [155](#)  
Cifra, 532–534

D

Red oscura , 5.559  
DAS, ver Almacenamiento conectado directamente  
Base de datos como servicio, [27](#), [30](#)  
Administrador de base de datos, [91](#)  
Administrador de búfer de base de datos, [24](#)  
Categorización de bases de datos, [587](#)  
Clúster de bases de datos, [384](#)  
Consistencia de la base de datos, [15](#), [110](#), [183](#), [185](#)  
Cracking de bases de datos, [81](#), [86](#)  
Integración de bases de datos, 5, [16](#), [25](#), [281](#), [285](#) binario,  
    [296](#) lógico, [282](#)  
    n-ario, [297](#) físico,  
    [282](#)

Integridad de la base de datos, [91](#)  
Replicación de bases de  
    datos véase replicación de datos, [247](#)  
Selección de bases de datos, [588](#)  
Servidor de base de datos, [21](#)  
Estadísticas de bases de datos, [159](#)  
Centro de datos, [29](#)  
Limpieza de datos, [306](#), [450](#), [589](#) nivel  
    de instancia, [306](#) nivel de  
    esquema, [306](#) web, [608](#)

Véase también Calidad de los datos

Control de datos, [14](#), [91](#)  
Diccionario de datos, [82](#)  
Directorio de datos, [82](#)  
Distribución de datos, [13](#)  
Cifrado de datos, [102](#)  
Fusión de datos, [610](#), [617](#)  
Guía de datos, [574](#)  
Independencia de datos, [3](#)  
    lógica, [8](#)  
    física, [8](#), [9](#)

Integración de datos, 5, [283](#), [512](#) web,  
    [588](#)  
    Véase también Integración de bases de datos  
Lago de datos, 5, [281](#), [332](#), [508](#), [515](#), [589](#), [608](#)  
Localidad de datos, [11](#), [33](#)  
Localización de datos, [33](#), [129](#), [136](#), [138](#), [140](#), [178](#), [333](#)

Registro de datos, [310](#), [311](#), [411](#)  
Particionamiento de datos, 8, [11](#), [34](#), [73](#), [80](#), [84](#), [244](#),  
    [353](#), [358–360](#), [390](#), [489](#), [537](#), [538](#), [606](#)

adaptativo, [78](#), [79](#)  
consciente de la carga de trabajo, [74](#), [78](#)  
Véase también Fragmentación

Encargado del [tratamiento](#) de datos, [24.354](#)  
Protección de datos, [102](#)  
Calidad de datos, [512](#)  
    web, [584](#), [608](#)  
    Véase también Limpieza de datos  
Replicación de datos, 8, [10](#), [15](#), [28](#), [29](#), [247](#), [454](#), [485](#)

ansiosa,  
    [394](#) centralizada,  
    [256](#) distribuida,  
    [262](#) copia primaria,  
    [260](#) maestro único, [256](#),  
    [258](#)  
actualización, [253](#) manejo  
de fallos, [272](#) comunicación de  
grupo, [269](#) centralizada  
diferida, [262](#) distribuida  
diferida, [268](#) copia primaria  
diferida, [265](#) maestro único  
diferido, [263](#), [265](#)  
actualización diferida,  
    [254](#) copia primaria, [255](#) maestro único, [254](#)

Sesgo de datos, [74](#), [375](#)  
Espacios de datos, [589](#)  
Flujo de datos, [470](#), [473](#), [513](#)  
Sistema de gestión de flujo de datos, [470](#), [471](#)  
Sistema de procesamiento de flujo de datos, [471](#)  
Sistema de flujo de datos, [471](#), [481](#), [485](#), [553](#)  
Traducción de datos, [304](#)  
Veracidad de los datos, [608](#)  
Almacén de datos, [282](#), [298](#), [306](#), [508](#)  
DBA, ver Administrador de base de datos  
DBPedia, [595](#)  
DB2 BigSQL, [510](#)  
Prevención  
    de bloqueos, [15](#)  
detección centralizada, [195](#)  
detección, [15](#)  
detección y resolución, [194](#)



Bloqueo muerto (cont.)  
    detección distribuida, 196 global, 194  
    detección  
        jerárquica, 195 prevención, 15 Árbol  
    de decisión, 295  
Web profunda, 559, 615  
Distribución de grados  
sesgados, 500 Conflicto de dependencia,  
289 Desconexión, 162 Relación  
diferencial, 97 DIKE,  
294, 335 DIPE, 335  
Almacenamiento  
conectado  
directamente, 358 Desunión, 37 Sistema  
de computación  
distribuida, 1 Control de concurrencia  
distribuida, 10, 14 Consenso distribuido, 231 Base de  
datos distribuida, 1 diseño, 13 sistema  
de gestión, 1 fiabilidad, 15  
    Bloqueo  
        muerto distribuido, 194 Directorio  
        distribuido, 91  
Monitor de ejecución distribuida, 24  
Consistencia de réplica de tabla  
hash distribuida, 428 Unión distribuida, 24  
Consulta distribuida, 34, 129  
    optimización dinámica, 130, 136  
ejecución, 136, 139 plan de  
ejecución, 136 optimización híbrida,  
130 procesamiento, 14 optimización, 139  
optimización estática, 130,  
165 Protocolos de  
recuperación distribuida, 10  
Confiabilidad  
distribuida, 10 Sistema  
de almacenamiento distribuido, 451  
Transacción distribuida, 34 registro, 235  
administrador, 24 Definición de tipo  
de documento, 595 Restricción de dominio,  
113 DSMS, consulte Sistema de  
    administración de  
flujo de datos DSPS, consulte Sistemas de  
procesamiento de flujo de datos  
DSS, consulte Sistema de flujo de datos DTD, consulte Definición  
de tipo de documento Optimización dinámica de consultas  
distribuidas, 161 Programación dinámica,  
135 DynamoDB, 444, 522–524, 554, 556

mi  
Eddy, 174–177, 179, 336  
BSP centrado en el bordeBloque de procesamiento  
    de gráficos  
        centrado en el borde sincrónico, 507  
Procesamiento de gráficos centrado en el borde,  
495 asíncrono, 507  
    síncrono de bloque, 507  
    recopilación-aplicación-dispersión, 507  
Distancia de edición, 292  
EBurro, 444  
Edutella, 407, 411, 445  
Elasticidad, 28  
Escalabilidad elástica, 455  
Correspondencia a nivel de elemento, 288, 289, 293  
Vinculación, 596  
Modelo de datos entidad-relación, 285  
Protocolo de epidemia, 401  
Modelo ER, 285, 337  
Esgyn, 537  
Estocada, 545, 548, 549, 554, 555  
Éter, 440, 442  
ETL, véase Extraer–transformar–cargar  
Semántica de exactamente una vez, 485  
Extraer-transformar-cargar, 282, 509, 528, 537, 544

F  
F1, 537, 538, 555, 556  
Conmutación por error, 352, 383, 384, 386, 453, 454  
Fallos, 15 de  
    comisión, 240 de  
    comunicación, 209 de  
    omisión, 240 de sitio,  
220 de  
    transparencia, 10  
Base de datos federada, 5  
Sistema de base de datos federada, 16  
Obtener según sea necesario, 166, 179  
Asignación de archivos, 67  
Almacenamiento de archivos, 452  
Arreglar/limpiar, 234  
Flink, 471, 514  
FlumeJava, 460, 513  
Flujo, 483, 514  
Restricción de clave externa, 112  
Adelante, 541, 543, 554, 556  
Fragmento, 13, 36, 37, 41–43, 45–53, 62–73, 78, 83, 84, 89  
  
Fragmentar y replicar, 163, 364

Fragmentación, 8, 13, 33, 35, 37, 41–43, 48–52, 54, 56, 65, 66, 68, 72, 73, 83, 84, 87, 89, 130, 134, 140, 142, 359 completitud, 36 derivada horizontal, 37 deriva, 130 disjunción, 36 hash, 73 horizontal, 35, 130 derivada, 48 primaria, 141 híbrida, 35, 130, 148 anidada, 35 predicado, 41, 143 primaria horizontal, 37, 40 rango, 74 reconstructibilidad, 36 round robin, 73 regla, 138, 141, 142 vertical, 35, 130, 143

Freenet, 399

Reductor completo, 154

Top-k totalmente descentralizado, 419

Completamente duplicado, consulte Base de datos completamente replicada

Base de datos completamente duplicada, consulte Base de datos completamente replicada

Base de datos completamente replicada, 13, 67

Restricción de dependencia funcional, 112

Mesa de fusión, 589

Recuperación de brecha, véase semántica de "como máximo una vez"

Ajo, 327

GAS, ver Recolectar-aplicar-dispersar

Recopilar-aplicar-dispersar, 495, 497, 504 asíncrono, 497 centrado en el vértice, 503

GAV, ver Global como vista

Restricción general, 112

DBMS geodistribuido, véase DBMS distribuido geográficamente

DBMS distribuido geográficamente, 2

GFS, ver Sistema de archivos de Google

GFS2, véase Sistema de archivos global 2

Gigascopio, 471, 514

Jirafa, véase Jirafa apache

Jirafa++, 505

GiraphUC, 502

Medida de afinidad global, 56

Vista global, 283, 303, 310–312, 334, 543, 554

Regla de confirmación global, 213

Esquema conceptual global, 23, 25, 33, 137, 281–283, 285, 287, 296–299, 302–304, 332, 337

Directorio/diccionario global, 83

Sistema de archivos global 2, 454

Índice global, 360

Global-local-como-vista, 283, 303, 304, 311

Optimización de consultas globales, 136

Optimizador de consultas global, 24

Esquema global, 410

Gráfico de espera global, 194, 195

Glosario, 588

PEGAMENTO, 410

GlusterFS, 454

Gnutella, 396, 399, 428, 444

Google Bigtable, 529–531, 541, 554, 556

Sistema de archivos de Google, 452–454, 529, 530

Lenguaje de consulta de Google, 541

Protocolo de chismes, 401

GPS, 498, 515

GQL, ver Lenguaje de consulta de Google

Análisis de gráficos, 451, 489

DBMS, 531

dirigido, 194, 294, 486, 532, 560 dirigido acíclico, 461, 569 etiquetado en el borde, 486, 572 Facebook, 486 Friendster, 487 ley de potencia, 486 red de carreteras, 487 sin escala, 486 Twitter, 486 no dirigido, 486 web, 486, 487, 560, 561, 564, 568, 616 ponderado, 486

GraphBase, 535

Isomorfismo de grafos, 487

GraphLab, 503, 504, 515

Partición de gráficos, 489 corte de arista, 489, 491 arista disjunta, 489 corte de vértice, 489 vértice disjunto, 489

Sistemas de grafos, 451

GraphX, 470, 494, 513, 515

Ganancia de cuadrícula, 537

Comunicación grupal, 269

Agrupación, 53

GSQL, 474, 513

Tienda G, 605, 606, 617

H

Hadoop, [86](#), [450](#), [458](#), [461](#) , [464](#), [494](#), [508–510](#), [531](#), [546](#), [547](#), [554](#)  
SQL, [510](#)  
HadoopDB, [545](#), [547](#), [554–556](#)  
Sistema de archivos distribuidos Hadoop, [454](#), [455](#), [458](#), [465–469](#), [508–510](#), [528](#), [531](#), [538–540](#), [544–550](#), [554](#), [555](#), [557](#), [606](#)

HaLoop, [494](#)  
Índice hash, [360](#)  
Partición hash, [481](#)  
Base de datos, [531](#)  
HDFS, consulte Sistema de archivos distribuido Hadoop  
HDInsight, consulte Azure HDInsight  
Latidos del corazón, [480](#)  
Garza, [471](#)  
Heterogeneidad, [19](#)  
Hexastore, [602](#), [617](#)  
Web oculta, [559](#), [584](#), [585](#), [615](#)  
Historia, [185](#), [188](#), [201](#), [204](#), [244](#), [250–252](#), [258](#), [264](#), [438](#)  
global, [188](#), [207](#), [251](#), [252](#), [255](#), [256](#), [258](#), [264](#), [265](#), [267](#), [272](#)  
local, [188](#), [208](#)

Algoritmo HITS, [568](#), [616](#)  
Colmena, [460](#)  
HiveQL, [460](#), [510](#), [513](#)  
Homónimo, [290](#), [291](#)  
Escalamiento horizontal, véase Arquitectura de escalamiento horizontal  
HTAP, [528](#), [537](#), [538](#), [556](#)  
HTML, [591](#)  
Hurón, [471](#), [482](#), [514](#)  
Emparejamiento híbrido, [295](#)  
Optimización de consultas híbridas, [136](#)  
distribuidas, [169](#)  
Hyperledger, [442](#)  
Tela, [442](#), [443](#)  
Iroha, [443](#)  
Hiperónimo, [290](#), [291](#)

I

IaaS, véase Infraestructura como servicio  
IBM DB2RDF, [602](#), [617](#)  
ICQ, [396](#)  
IMAP, [296](#)  
Dependencia de inclusión, [110](#)  
Paralelismo independiente, [13](#)  
Protocolo de recuperación independiente, [211](#), [220](#)  
Restricción individual, [117](#), [119](#)  
Gráfico infinito, [535](#)  
Integración de información, [283](#)  
Infraestructura como servicio, [5](#), [27](#), [30](#)

INGRESOS, [4](#), [31](#), [92](#), [123](#), [452](#)  
distribuidos, [161](#), [179](#), [261](#), [278](#)  
Coincidencia basada en instancias, [289](#), [291](#)  
Coincidencia de instancias, [288](#) integración, [285](#), [296](#)  
Restricción de integridad, [183](#)  
Internet de las cosas, [442](#), [445](#)  
Interoperabilidad, [281](#)  
Equilibrio de carga entre operadores, [378](#)  
Paralelismo entre operadores, [11](#), [12](#), [369](#)  
Paralelismo entre consultas, [11](#), [33](#)  
Reglas interesquemáticas, [291](#)  
Equilibrio de carga intraoperador, [376](#)  
Paralelismo intraoperador, [11](#), [363](#), [369](#)  
Equilibrio de carga entre consultas, [378](#)  
Paralelismo intraconsulta, [11](#), [33](#), [36](#)  
Reglas intraesquemáticas, [291](#)  
Algoritmo de regla inversa, [315](#)  
Costo de E/S, [134](#)  
IoT, ver Internet de las cosas  
Nivel de aislamiento, [188](#), [189](#), [203](#), [237](#), [252](#), [254](#)

Yo

JAQL, [460](#), [513](#)  
JDBC/ODBC, [460](#)  
JEN, [554](#), [556](#)  
Jena, [602](#), [604](#), [617](#)  
Rastreador de empleo, [459](#)  
Unir gráfico, [37](#), [40](#), [47–49](#), [52](#), [87](#), [138](#), [152–156](#), [176](#), [181](#), [182](#) particionado, [49](#)  
simple, [49](#), [52](#)  
Implementación de unión en MapReduce, [461](#)  
Ordenamiento de unión, [139](#), [149](#), [151](#) distribuido, [130](#), [149](#), [178](#)  
Únete al trie, [149](#)  
JSON, [525](#), [528](#), [538](#), [543](#), [556](#) binario, [525](#)  
JXTA, [407](#), [445](#)

K

Kazaa, [396](#), [399](#), [428](#)  
Conflicto clave, [289](#)  
División de claves, [482](#)  
Almacén de clave-valor, [521](#)  
KIVI,  
algoritmo de [539](#) k-medias, [467](#)

Yo

Propagación de etiquetas, [491](#)  
Latencia, [11](#)

LAV, ver Local como vista

LCS, véase Esquema conceptual local

LeanXcale, 237, 537, 538, 540, 555, 556

Emparejamiento basado en el aprendizaje, 294

Trie profundo izquierdo, 371

Trie de unión lineal izquierda, 150

Métrica de Lewenstein, 292

LFGraph, 498, 515

Trie de unión lineal, 150

Coincidencia lingüística, 291

Análisis de enlaces, 567

Datos abiertos vinculados, 589, 590, 595, 607, 617

Equilibrio de carga, 374

Local como vista, 283, 310, 311, 314, 334, 410, 541, 554

Esquema conceptual local, 23, 281–283, 285, 287, 296, 298, 299, 303, 304, 306, 332

Directorio/diccionario local, 83

Consulta local, 139

Optimizador de consultas local, 24

Gerente de recuperación local, 24, 186

Modelo relacional local, 409

Gráfico de espera local, 194, 195

Bloqueo, 15

Modo de bloqueo, 189

Mesa de cerradura, 189

LOD, ver Datos Abiertos Vinculados

Registro, 186 estable, 187 volátil, 186

Lorel, 573, 616

Descomposición sin pérdidas, 36

LRM, ver Modelo Relacional Local

LSD, 295, 296

Lucene, 536

METRO

Aprendizaje automático, 450, 487

Función de mapa, 455

Unión solo al mapa, 463

Creación de mapas, 298

Mantenimiento de mapas, 298, 304

MapReduce, 451, 455–464, 466–468, 470, 494, 498, 509, 512, 513, 515, 528, 544–549, 554, 555, 606, 615

Mezcla, 589

Sitio maestro, 161, 165, 254

Programa de materialización, 138

Vista materializada, 96, 98, 100, 102, 122–124, 315, 474, 545, 548, 549, 556 mantenimiento, 96, 124, 282

Maveric, 305

Consulta de contenido máximo, 315

MDBS, véase Sistema multibase de datos

Esquema mediado, 25, 281, 282, 285, 298

Mediador, 25, 309

Arquitectura de mediador/envoltorio, 25, 309, 310, 323, 333, 335

Memcached, 524

MemSQL, 537

Metadatos, 83

Metabuscador, 569, 586, 588, 615

METIS, 490–492

Rueda de molino, 471, 485

Algoritmo MinCon, 315, 316

Fragmento de minterm, 42, 45

Predicado de mintérmino, 39–47, 52, 53

Selectividad de minterm, 40

MISO, 556

Fragmentación mixta, 66

Mizan, 498, 515

MonetDB, 36

MongoDB, 525–527, 554, 556

Consulta monótona, 474

Mulder, 580, 616

Multibase de datos, 5, 17, 318, 326

optimización de consultas, 317

procesamiento de consultas, 307 sistema, 16, 25, 31, 281, 283, 307–309, 324, 328, 337, 346

Multiinquilino, 30

Multigrafo, 486

Optimización de múltiples consultas, 480

Control de concurrencia multiversión, 203

Coherencia mutua, 14

MVCC, consulte Control de concurrencia multiversión

NOTA

NAS, ver Almacenamiento conectado a red

Tupla negativa, 478

Neo4j, 532–535, 554, 556

Fragmentación anidada, 66

Unión de bucle anidado, 475

Almacenamiento conectado a red, 357

Sistema de archivos de red, 357

Particionado de red, 15, 210, 227 múltiple, 228 simple, 228

Red neuronal, 295

NewSQL, 3, 30, 519–521, 535, 537, 538, 554–557

NFS, ver Sistema de archivos de red

N-grama, 292

Sin arreglo/sin descarga, 234

nonce, 441

Estado no comprometido, 226

N1QL, 522

Restricción de atributo no nulo, 112

Base de datos no replicada, 13, 67

SQL sin interrupciones, 194

Arquitectura de memoria no uniforme, 356, 391

    caché coherente, 356

NoSQL, 2, 3, 16, 17, 20, 30, 359, 450, 451, 455, 511, 519–521, 525, 528, 536–540, 543, 544, 551, 553–557 multimodelo, 535, 536

NuoDB, 537

Partición de N vías, 63

Oh

Modelo de intercambio de objetos, 570, 616

Almacenamiento de objetos, 454

Almacén de objetos, 452

OceanStore, 432

Odisea, 554, 556

OEM, véase Modelo de intercambio de objetos

OLAP, ver Procesamiento analítico en línea

OLTP, ver Procesamiento de transacciones en línea

Equivalencia de una copia, 247, 250

Serialización de una sola copia, 252

    fuertes, 254

Procesamiento analítico en línea, 96, 184, 244, 282, 333, 349, 359, 387, 388, 390, 392, 508, 509, 511, 519, 520, 537–539, 545

Consulta de gráficos en línea, 487

Carga de trabajo de gráficos en línea, 487

Procesamiento de transacciones en línea, 184, 244, 349, 357, 359, 390, 470, 508, 537, 538

Ontología, 290

Operador trie, 149, 369

Control de concurrencia optimista, 15

Oracle NoSQL, 524

OrientDB, 536, 554

Red superpuesta, 398, 405 puro, 398

PAG

PaaS, ver Plataforma como servicio

PageRank, 466, 487, 488, 495, 504, 563, 564, 567, 616

Arquitectura paralela, 352

Unión asociativa paralela, 424

DBMS paralelo, 3, 16

Unión hash paralela, 364, 366, 424

Combinación y ordenación en paralelo, 364

Unión de bucles anidados paralelos, 364

Optimización de consultas paralelas, 369

Evaluación de funciones parciales, 606

Agrupación parcial de claves, 482, 514

Base de datos parcialmente duplicada, consulte Base de datos parcialmente replicada

Base de datos parcialmente replicada, 13, 67, 95

Tiempo de espera del participante, 221

Enfoque centrado en la partición, 495

Procesamiento de gráficos centrado en

    particiones asíncrono, 506

    bloque síncrono, 504 recopilación-aplicación-dispersión, 506

Base de datos particionada, 13, 67

Particionado, 33, 62

Expresión de ruta, 573

Paxos, 231, 441

    básico, 232

Integración de pago por uso, 589

Base de datos de pares, 411

Computación peer-to-peer,

    17, 395, 16, gestión de datos, 395

    DBMS, 22

    jerárquico estructurado, 445 puro, 398

    replicación, 428

    estructurado, 398, 402, 403, 425, 434, 447

    superpeer, 406, 444, 447

    sistemas, 4, 17, 19, 23, 24, 288, 395–399, 401–403, 405, 406, 408–412, 419, 421, 425, 426, 428, 429, 431, 432, 436–438, 444–448, 522 no estructurado, 398, 399, 402, 411, 419, 444–448

Pentaho, 510

Entrega periódica de datos, 6

Consulta persistente, 471

Control de concurrencia pesimista, 15

PGrid, 405, 409, 432, 434, 445, 446

Fantasma, 245

PHJ, ver Unión hash paralela

FORIZONTAL, 44

PHT, 405

Plaza, 409

MUELLE, 424

PIERjoin, 424

Latín de cerdo, 460, 513

Unión hash simétrica canalizada, 476

Paralelismo de tuberías, 12

PIW, ver Web indexable públicamente

PKG, consulte Agrupación de claves parciales

PlanetP, 419

Función de planificación, 326

Plataforma, 510

Plataforma como servicio, 5, 27, 29, 30

PNL, consulte Unión de bucles anidados paralelos

Polibase, 545–547, 554–556

Polystore, 519, 520, 538–540, 544, 548, 553–556

    híbrido, 549, 556

    débilmente acoplado, 540, 556

    fuertemente acoplado, 544, 556

Prueba posterior, 114

PoW, véase Prueba de trabajo

Power BI, 510

PowerLyra, 493

Recuperación precisa, véase semántica exactamente una vez

Restricción de condición previa, 112

Restricción predefinida, 112

Prefijo hash trie, 426

Pregel, 498, 515

Pregelix, 498, 515

Prueba previa, 114, 115

Prueba de trabajo, 441

Gráfico de propiedades, 486

Tabla de propiedades, 602, 617

P2P, ver sistema peer-to-peer

Web indexable públicamente, 559, 584

Sistema de publicación/suscripción, 473

Puntuación, 476, 513

Sistema basado en push, 5, 6

Q

QoX, 541, 544, 554, 556

Consulta

    algebraica, 129, 137–140

    descomposición, 136, 137

    distribuida, 95

    ejecución, 311, 329 plan

    de ejecución, 133 gráfico, 137

    modificación, 93

    optimización, 129

        dinámico, 136, 165

        estático, 136

    procesamiento, 129

    procesador, 129, 354

    reescritura, 310

        uso de vistas, 315

    traducción, 311, 329

Sistema de preguntas y respuestas, 580

Quórum, 230

Protocolo de votación basado en quórum, 275

R

R\*, 179, 194

Balsa, 556

Partición de rango, 530

Consulta de rango en sistemas P2P, 425

Clasificación, 563, 564, 567

RavenDB, 528

RDD, véase Conjunto de datos distribuidos resilientes

RDF, véase Marco de descripción de recursos

RDF-3X, 602, 617

Consulta de accesibilidad, 487

Protocolo disponible de lectura única/escritura total, 273–276, 278

    distribuido, 274

Protocolo de lectura única/escritura total, 253, 259, 262, 273–275

Cuórum de lectura, 275

Reconstrucción, 37

Recuperación, 15, 183

    protocolo, 211, 224

Redis, 524

Función de reducción, 455

Reductor, 151, 153

Técnica de reducción, 140

Borde referencial, 294

Integridad referencial, 51

Predicado simple relevante, 43

Confiabilidad, 10, 15

Repartición de unión, 463

Base de datos replicada, 13

Replicación

    P2P, 247, 428

    Véase también Replicación de datos

Resiliencia, 183

Conjunto de datos distribuidos resilientes, 468, 469, 494

Marco de descripción de recursos, 444, 486, 595, 598, 615, 617 gráfico, 486, 597–599, 605, 606 esquema, 591, 596

Tiempo de respuesta, 134, 157, 158

Tupla de revisión, 473

Riak, 524

Trie profundo derecho, 371

Recuperación de reversión, consulte Semántica de "al menos una vez"

Partición round-robin, 481

ROWA, ver Protocolo de lectura única/escritura total

ROWA-A, consulte Protocolo disponible de lectura única/escritura total

Procesador de soporte en tiempo de ejecución, 24

S

SaaS, ver Software como servicio

SAN, ver Red de área de almacenamiento

SAP HANA, 537

Diente de sierra, 443

Sawzall, 460, 513

Arquitectura de escalamiento horizontal, 13	
Ampliación, 11	
Programador, 186	
Esquema, 2	
adaptación, 304	
generación, 283	
heterogeneidad, 287, 289	
integración	
binaria, 296	
nario, 296	
mapeo, 285, 287, 298	
coincidencia, 285, 287, 288	
traducción, 283, 345	
Coincidencia basada en esquemas, 289, 291	
Coincidencia a nivel de esquema, 293	
Esquema en lectura, 508, 509	
Esquema en escritura, 508	
Cisma, 74, 85	
SDD-1, 179	
Motor de búsqueda, 560, 562	
Espacio de búsqueda, 133, 369	
Estrategia de búsqueda, 135, 373	
Factor de selectividad, 160	
Control de	
datos semánticos,	
14 controlador de	
datos, 24	
heterogeneidad, 290 restricción	
de integridad, 91, 110 control de	
integridad, 91, 92,	
110 traducción, 304 web, 409, 590, 595, 617	
Sistema semiautónomo, 18	
Semiunión,	
programa 153 , 154–156, 179	
SEMINT, 295	
Datos semiestructurados, 569, 570	
Serializabilidad, 185, 188 una	
copia, 252	
Acuerdo de nivel de servicio, 28	
Arquitectura orientada a servicios, 27	
Sésamo, 601, 617	
SETI@casa, 396	
Restricción orientada a conjuntos, 117, 118, 120	
Fragmentación, 36	
Disco compartido, 357	
Memoria compartida, 355	
Nada compartido, 358	
Barco entero, 166	
Barajar, 457	
Particiones aleatorias, 481	
Si, véase Aislamiento de instantáneas	
Inundación por similitud, 335	
Valor de similitud, 287	
Predicado simple, 38, 39, 42–46, 53	
completitud, 42	
minimalidad, 42	
Particionamiento virtual simple, 387	
Simplificación, 115	
DBMS de ubicación única, 2	
Ruta más corta de una sola fuente, 487	
Gráfico de salto, 406	
SkipNet, 406	
Sitio de esclavos, 254	
Base de datos de instantáneas, 470	
Aislamiento de instantáneas, 184, 185, 189, 203, 207–	
209, 236–239, 242, 243, 254, 278 fuerte, 254	
Software como servicio, 5, 27, 29, 30	
Combinación de ordenamiento y fusión, 365	
Código Soundex, 292	
Precisión de la fuente, 612	
Dependencia de la fuente, 613	
Frescura de la fuente, 614	
Esquema fuente, 285	
Llave inglesa, 237, 538	
Chispa, 451, 455, 466–470, 487, 494, 509, 511–513,	
528, 545, 549–552, 554, 555, 617	
Sparksee, 535	
Spark SQL, 554	
SPARQL, 570, 598–602, 604–607, 617 distribuido,	
606 punto final, 607	
Máquina de empalme, 537	
División, 53	
SQL++, 528, 543, 556	
Sistema de arranque, 580, 616	
Optimización estática, 139	
Red de área de almacenamiento, 358	
ARROYO, 471, 514	
Datos de transmisión, 451	
StreaQuel, 474, 513	
Consulta de componente fuertemente conectado, 489	
Conflicto estructural, 289	
Restricción estructural, 110	
Similitud estructural, 293	
Emparejamiento basado en la estructura, 293	
Coincidencia a nivel de estructura, 288, 289	
Índice de estructura, 566	
StruQL, 575, 616	
Coincidencia de subgrafos, 487	
Sistema superpares, 398	
Superpaso, 496	
SVP, consulte Particionado virtual simple	
ESPADA, 76, 80, 85	
Tienda SW, 617	
Unión hash simétrica, 368	

Sinónimos, 290, 291	
SystemML, 460, 513	
Sistema R, 92	
Sistema R*, 194, 196	
T	
TA, ver algoritmo de umbral	
Cuadro, 510	
Tableta, 530	
Tapiz, 404, 429	
Esquema de destino, 285	
TelegraphCQ, 471, 514	
Tenzing, 460, 513	
Protocolo de terminación, 211, 220 sin	
bloqueo, 211, 220	
Índice de texto, 566	
Piensa como un vértice, 495	
Compromiso trifásico, 226	
Algoritmo de umbral uniforme trifásico, 415	
Algoritmo de umbral, 412, 413	
Integración estrecha, 18	
Tiempo de espera, 210	
Marca de tiempo, 197, 198, 200, 202, 203, 206 orden,	
189	
ordenamiento, 197, 202	
básico, 198	
conservador, 201–203 lectura,	
198	
escritura, 198	
Sellado de tiempo, 15	
Flujo de tiempo, 471	
Consulta sobre viajes en el tiempo, 203	
Titán, 535	
Diseño de bases de datos de arriba hacia abajo, 13	
Consulta top-k, 412	
Optimización de costes totales, 134	
Aislamiento total, 19	
Tiempo total, 157	
TPUT, ver Umbral uniforme trifásico	
Algoritmo	
Transacción, 183	
atomicidad, 185	
conjunto base,	
185 anidado cerrado,	
240 consistencia, 183, 249, 251, 252, 277	
distribuido, 10	
durabilidad, 185, 556	
plano, 239	
inversión, 254	
aislamiento, 185	
administrador de	
registro (ver Registro), 186	
anidado, 240	
abierto anidado, 240	
conjunto de lectura,	
185 actualizar,	
254 dividido,	
240 conjunto de escritura, 185	
Restricción de transición, 113	
Transparencia, 1, 7, 21	
concurrency, 10	
distribución, 9	
fallo, 10	
fragmentación, 9	
ubicación, 9	
denominación,	
9 red, 9	
replicación, 10	
Consulta de árbol, 155	
Tribeca, 473, 514	
Trinidad, 498, 515, 535	
Trito, 580, 616	
Confirmación de dos fases, 10, 211, 243	
centralizada, 213	
distribuida, 215	
lineal, 214	
anidada, 214	
aborto presunto, 218	
confirmación presunta, 219	
Bloqueo de dos fases, 189	
centralizado, 190	
distribuido, 193	
copia primaria, 261 sitio	
primario, 190 estricto,	
201	
Conflicto de tipos, 289	
Tú	
UDF, ver Función definida por el usuario	
UMA, 355	
Despliegue, 313	
Acceso uniforme a la memoria, 355	
Aborto unilateral, 212	
Uniprot RDF, 595	
Restricción de clave única, 112	
UnQL, 616	
Función definida por el usuario, 457	
Controlador de interfaz de usuario, 24	
Procesador de usuario, 24	
V	
Árbol VBI, 406	
Veracidad, 450, 512	
véase también Calidad de los datos, 512	



Procesamiento de gráficos centrado en vértices,  
495 asíncrono, 501  
síncrono de bloques, 498, 499  
recopilación-aplicación-dispersión, 503  
Vertical, 36  
Vista, 91, 92, 311, 313  
definición, 93, 310  
gestión, 91, 92  
materialización, 92  
materializado, 92  
Máquinas virtuales, 28  
Relación virtual, 92  
VoltDB, 537  
Protocolo basado en votación, 230

O

CMI, 501, 505  
Componente débilmente conectado, 489, 499

Rastreo web, 563

fusión de datos,  
610 gestión de datos, 559  
gráfico, 560  
indexación, 616  
portal, 589  
consultas, 569  
búsqueda,  
562 servicio,  
27 tabla, 589

Registro web, 575, 616  
WebOQL, 575, 577, 616

Control de calidad web, 580, 616

WebSQL, 575, 577, 616

Tienda de columnas anchas, 529

Ventana, 471, 473–476

agregado, 479  
basado en conteo, 473, 478  
fijo, 473

unirse, 476, 478  
punto de referencia,  
473 modelo,  
473 particionado,  
473 predicado,  
473 consulta, 472,  
474 sesión, 474  
deslizante, 473  
basado en tiempo, 473,  
478 definido por el usuario, 474

Ejecución en ventana, 471, 474, 476, 477

Flujo de trabajo, 240

World Wide Web, 4, 16, 559

Envoltorio, 298, 309

Esquema de envoltura, 311

Cuórum de escritura, 275

W3QL, 575, 616

WWW, ver World Wide Web

X

xLM, 544  
XML, 444, 525, 544, 547, 560, 591, 615  
documento trie, 593  
Esquema XML, 595  
Gráfico de esquema XML, 595  
XPath, 595  
XQuery, 595  
X-Stream, 507, 515

Y

Yago, 595  
YAML, 525

Z

Zigzag trie, 371  
Cuidador del zoológico, 511