

Medios de consulta de bases de datos orientados a usuarios finales

Edwin Gamboa

Maestría en Ingeniería, Facultad de Ingeniería

Universidad del Valle

Cali, Colombia

Email: edwin.gamboa@correounivalle.edu.co

Resumen—La experiencia de usuario y la usabilidad de los medios de consulta de bases de datos se ven afectadas por la disparidad entre los modelos mentales de los usuarios y los esquemas lógicos que soportan las bases de datos. En este artículo se presentan cinco enfoques de medios de consulta que busca resolver la disparidad antes mencionada. Los enfoques de consulta presentados son: visual, basado en lenguaje natural, conceptual, variante de SQL y basado en palabras clave. Para cada enfoque se revisaron un conjunto de trabajos que permitieran extraer sus principales características, ventajas y desventajas. Esta revisión permitió concluir que ninguno de los enfoques mejora a los medios de consulta actuales en términos de usabilidad y experiencia de usuario, impidiendo su uso masificado en el mercado actual.

Keywords—Lenguajes de consulta, enfoques de consulta, bases de datos, usabilidad, experiencia de usuario

I. INTRODUCCIÓN

El crecimiento y uso de la información se ha hecho evidente de forma importante en los últimos años. Esto se debe, entre otras cosas, al fácil acceso que actualmente tienen las personas a tecnologías como Internet. Este fácil acceso hace los medios de consulta de bases de datos puedan ser accedidos y consultados por todo tipo de usuarios, incluidos aquellos considerados como novatos en las tecnologías de la información [1].

De acuerdo con la ISO 9241-11 la usabilidad es la medida con la que un producto, sistema o servicio puede ser usado por un grupo específico de usuarios para lograr objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso específico. La efectividad se refiere a la precisión y completitud con la que los usuarios obtienen los resultados. La eficiencia se refiere a los recursos usados en relación con la efectividad obtenida. Por último, la satisfacción se refiere a la disposición positiva del usuario respecto al producto, sistema o servicio.

Para el caso de los objetivos de los usuarios que deseen consultar bases de datos, sus objetivos se pueden representar en términos de los siguientes requerimientos:

- Recuperar información en un tiempo corto [2].
- Un medio de consulta fácil de usar y entender [2].
- Recuperar resultados precisos y completos [3].
- Representación multidimensional de los resultados, tales como tablas o grafos [3].

Adicionalmente, las ISO 9241-210 define la experiencia de usuario como las percepciones y respuestas de los usuarios que

resultan de el uso o el uso anticipado de un producto, sistema o servicio.

A pesar de que los lenguajes de consulta tradicionales, como SQL, cuentan con un poder expresivo alto y son considerados robustos [4]; estos no cumplen con los requerimientos mencionados anteriormente. Esto se debe principalmente a que la gran mayoría de usuarios no son expertos en el dominio de las bases de datos [5].

Para consultar una base de datos, los usuarios deben lidiar con la complejidad de los Sistemas de Gestión de Bases de Datos. Estos sistemas, no están diseñados para los usuarios actuales, pues sus fundamentos fueron concebidos muchos años atrás [6]. Por lo tanto, dichos sistemas presentan una disparidad entre el modelo mental de los usuarios y la estructura real de los datos en las bases de datos [7]. Esta disparidad se debe principalmente a que las bases de datos cuentan con un esquema lógico sobre el cual basan su funcionamiento. Además, el buen diseño de bases de datos implica usar estrategias que permitan evitar problemas futuros en la misma. Una base de datos mal diseñada puede, por ejemplo, generar problemas redundancia de datos. Una de las estrategias de diseño más usadas para combatir estos problemas es la normalización. Sin embargo, esta estrategia resulta en esquemas más complejos y más lejanos a los modelos mentales de los usuarios [4].

Por lo anterior, si un usuario necesita consultar una base de datos, debe comprender el esquema lógico detrás de la misma [3]. Puesto que cuando realiza una consulta, el usuario debe conocer los nombres exactos de las relaciones y atributos que intervienen en la consulta. Además, el usuario debe ser capaz de identificar relaciones no directamente involucradas en la consulta, por ejemplo al expresar un *join* en una consulta SQL [4], [3]. En consecuencia, el proceso de consultar una base de datos es difícil, incluso para usuarios expertos [4], [3], lo cual representa dos fuentes de frustración para los usuarios: primero, la imposibilidad de recuperar información de la forma deseada y segundo la obtención de resultados inesperados cuando la consulta no ha sido formulada de forma adecuada [3].

Es decir que estos medios de consulta presentan problemas de usabilidad y afectan la experiencia de los usuarios negativamente.

Por lo tanto, los medios de consulta tradicionales representan un impedimento para los usuarios novatos que deseen

Tabla I
ENFOQUES DE CONSULTA DE BASES DE DATOS USABLES

Enfoque	Descripción	Referencias
Visual	Representa de forma visual las consultas y los resultados recuperados por las mismas.	[1], [8], [9], [10], [5], [11]
Basado en lenguaje natural	Permite al usuario formular consultas escribiendo en su propio lenguaje, por ejemplo inglés.	[2], [12], [7]
Conceptual	Permite al usuario formular consultas imprecisas, usa conceptos similares a los proporcionados por el usuario para determinar la consulta requerida para obtener los resultados esperados.	[13], [6]
Variante SQL	Lenguajes que conservan las ventajas y sintaxis de SQL, añadiendo variaciones para hacerlo mas amigable.	[4]
Basado en palabras clave	Recupera resultados basado en palabras clave proporcionadas por el usuario. Supone tolerancia de los usuarios a datos irrelevantes y a ausencias de datos relevantes.	[14]

consultarlas o accederlas; haciendo necesaria la concepción de enfoque diferentes a los tradicionales o la adición de modelos a más alto nivel [3].

Este artículo presenta un conjunto de enfoques orientados a resolver la disparidad entre los modelos mentales de los usuarios finales y los esquemas lógicos de las bases de datos. El artículo se encuentra organizado de la siguiente manera: primero, en la Sección II se describen los enfoques mencionados, a continuación en la Sección III se presenta un análisis general de los enfoques revisados y finalmente en la Sección IV se presentan algunas conclusiones derivadas de este trabajo.

II. ENFOQUES

Con el propósito mejorar la usabilidad de los medios consulta y mejorar la experiencia de usuario novatos y expertos, es necesario resolver la disparidad entre los modelos mentales de los usuarios y los esquemas lógicos de las bases de datos. Diferentes enfoques han sido propuestos, dichos enfoques comprenden la construcción de lenguajes, interfaces o incluso sistemas completos. En su mayoría representan una capa a más alto nivel que traduce consultas amigables al usuario en consultas en lenguajes de más bajo nivel como SQL. Vale la pena destacar que todos estas propuestas están orientadas a la consulta de datos estructurados. A continuación se presentan cinco de estos enfoques y en la Tabla I se presenta un resumen de los mismos.

II-A. Visual

El enfoque visual busca acercarse al modelo mental de los usuarios con el uso de representaciones gráficas de los elementos involucrados en una consulta. Este enfoque está orientado a usuarios novatos. Algunas de sus ventajas son:

1. Los usuarios pueden crear selecciones visuales de forma instantánea [5].
2. Comunican claramente la información contenida en las bases de datos [5].
3. Las consultas se derivan de inferencias perceptivas, en lugar de comparaciones cognitivas arduas [5].
4. Se orienta a usuarios novatos [10], [5].
5. Ofrece un nivel de abstracción alto [10]

Por otra parte, las principales desventajas de este enfoque son:

1. La mayoría de propuestas pertenecientes a este enfoque carecen de formalismo en su definición [5].
2. La mayoría de de propuestas pertenecientes a este enfoque tienen un poder de expresividad bajo [5].
3. Complejidad de implementación superior a propuestas que procesan consultas en texto simple[9].

En [5] se realiza una revisión de 80 sistemas visuales de consultas desde el año 1975 hasta el año 1996. Un sistema visual de consulta se define como un sistema de consulta que representa el dominio de interés y sus consultas relacionadas de forma visual. En este trabajo se proponen dos clasificaciones para estos sistemas. La primera se basa en la representación de la consulta. La segunda clasificación es basada en la representación de los resultados de la consulta. Ambas clasificaciones están compuestas por las siguientes clases: basados en formularios, basados en diagramas, basados en iconos e híbridos. Además, en este trabajo se presenta una clasificación de usuarios y la relaciona con las clases antes mencionadas. En este trabajo se presentan algunas evidencias de que estos sistemas presentan un desempeño y usabilidad superior a la de SQL, cuando de realizar consultas se trata.

En [1] se presenta una extensión de la revisión descrita anteriormente. En este trabajo se incluyen 34 nuevos sistemas visuales de consulta hasta el año 2016. Este trabajo adiciona dos clases a la clasificación presentada en [5], estas son por facetas y desconocido. Adicionalmente, en este artículo se concluye que este tipo de sistemas parecen no ser efectivos, pues la mayoría son prototipos que no se usan en ambientes reales o representan herramientas comerciales.

Algunos trabajos no incluidos en los documentos anteriores se presentan a continuación:

En [8] se propone un lenguaje visual de consulta construido usando un enfoque centrado en el usuario. Este lenguaje permite al usuario construir consultas manipulando una representación visual del esquema lógico de la base de datos. Además, ofrece ayudas visuales que orientan al usuario en el proceso de filtra de datos y construcción de rutas de las consultas (por ejemplo *joins*). En este trabajo se presentan una evaluación en la que se probaron dos aspectos: la efectividad de las consultas realizadas y usabilidad. En ambos aspectos se lograron resultados positivos. Sin embargo, no se describen detalles importantes como el número de participantes de la evaluación. Finalmente, el hecho de que para construir una consulta se parte de una representación visual del esquema

lógico, esto hace que la efectividad se pueda ver afectada cuando dicho esquema sea demasiado complejo.

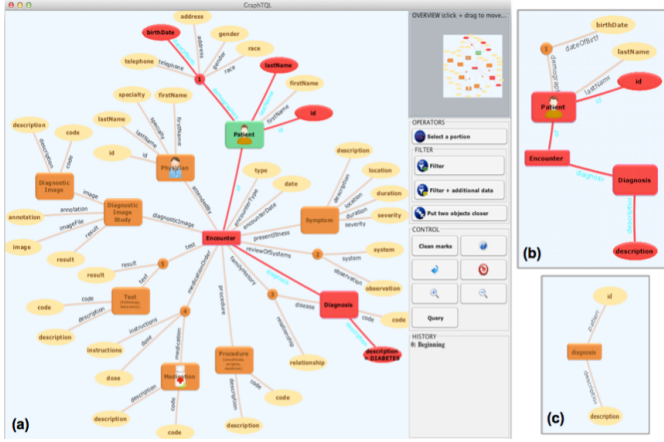


Figura 1. Ejemplo de representación de consulta para [8], a) Muestra un ejemplo esquema lógico y la interfaz de construcción de consultas b) y c) son ejemplos de consultas

En [9] se propone un lenguaje visual para consultas *ad-hoc*. Este presenta una descripción detallada de la sintaxis del lenguaje, la cuál se basa en la notación UML. Un ejemplo de una consulta se presenta en la Figura 2 También se presenta la arquitectura del sistema que lo soporta. Para el ese momento, el sistema era aún un prototipo, que aún no soportaba todas las funcionalidades planteadas por el autor. Para este trabajo se realizaron dos pruebas de usabilidad, una exploratoria y una final. Se obtuvieron resultados positivos comparado con otro lenguaje visual. En las prueba final participaron 39 personas provenientes del área de las tecnologías de la información. En el artículo se presentan resultados de expresividad, genericidad, usabilidad, practicabilidad y rendimiento. En general este presenta buenos resultados, pero en todos los aspectos es superado por al menos otro enfoque visual. Una desventaja de este lenguaje puede ser el hecho de que se basa en UML, pues esto representa el aprendizaje por parte del usuario final de una notación que puede resultar compleja dependiendo de su nivel de experticia.

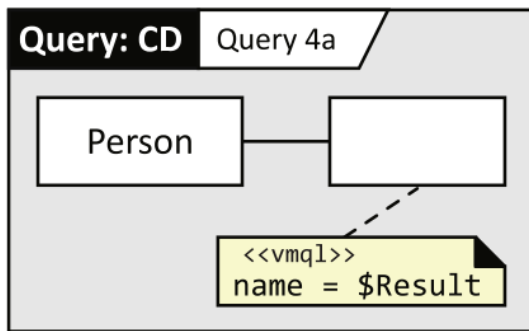


Figura 2. Ejemplo de consulta para [9]

En [11] se presenta un enfoque visual de consulta de bases

de datos orientado al usuario final. El enfoque es soportado a través de una aplicación web que permite representar las consultas en forma de árbol con ayuda del lenguajes XML y obtener resultados de la base de datos a través de archivos JSON. El usuario construye sus consultas en forma de árbol y el sistema presenta resultados inmediatos, lo cual puede ayudar al usuario a detectar errores en tiempo real. Su poder de expresión incluye selecciones, proyecciones, unión, intersección y operaciones *in* y *not in*. En la Figura 3 se presenta un ejemplo de una consulta.

Se presenta una evaluación de usabilidad en la que participaron 12 usuarios con conocimiento del dominio de aplicación, se obtuvieron resultados positivos en términos de la correctitud de las consultas.

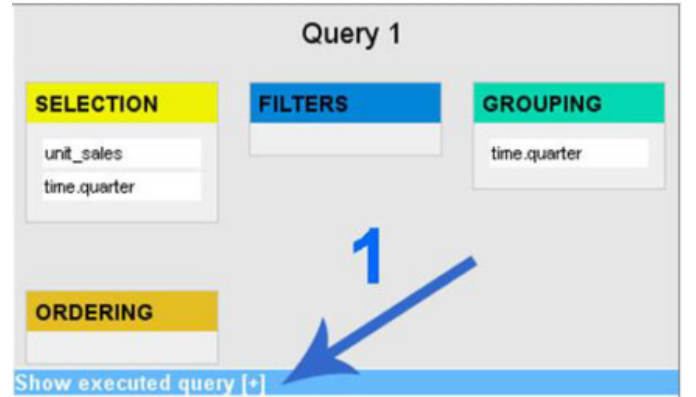


Figura 3. Ejemplo de consulta para [11]

En [10] se propone un lenguaje de consulta objeto-relacional visual, que ofrece independencia física y lógica. Este lenguaje se basa en iconos (principalmente botones y flechas), menús y formularios visuales. Además, guía al usuario en diferentes pasos para construir una consulta. Estos pasos son: duplicar, restringir, acción, condición y formato. Una vez el usuario ha construido su consulta, esta es transformada a una equivalente en lenguaje SQL. Un ejemplo de una consulta de actualización se presenta en Figura 4 y en Figura 5 se presenta la interfaz gráfica que permite construir las consultas.

Este trabajo extiende el enfoque de *Query by Example (QBE)* de dos formas. En primer lugar, se permite unir dos relaciones en un solo formulario (equivalente a las tablas en QBE) y en segundo lugar, se pueden crear diferentes tipos de formularios dependiendo del paso en el que se encuentre el usuario. Este lenguaje, tiene un poder expresivo igual al de SQL o QBE, pues cumple con el criterio de completitud relacional. Incluyendo la posibilidad de modificar la base de datos. Además, este modelo ofrece a usuarios avanzados la posibilidad de usar consultas en SQL.

II-B. Basados en lenguaje de natural

Este enfoque busca que los usuarios puedan realizar consultas a la base datos utilizando lenguaje natural, por ejemplo

Insert Form				
Supplier				
	Supp#	SName	Status	SCity
	=50	='Elite Distributor'	=5	='New York'

Figura 4. Ejemplo de consulta para [10]

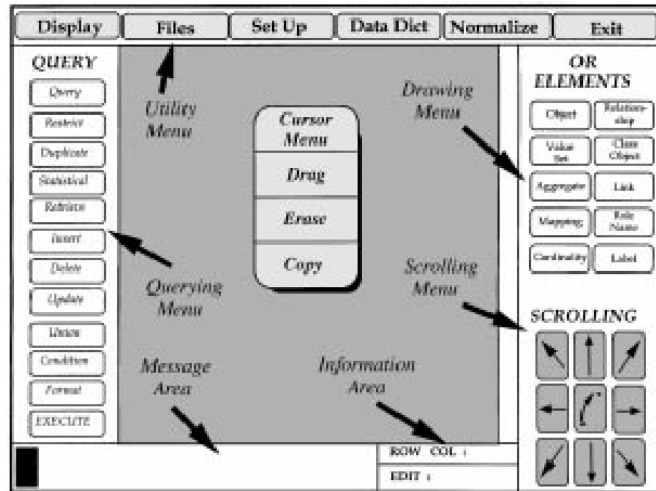


Figura 5. Ejemplo de consulta para [?]

escribir consultas usando gramática del inglés. Las principales ventajas de este enfoque son:

1. Naturalidad y facilidad de uso para el usuario, debido a que su forma de interacción es más cercana a la comprensión humana [2].
2. Mejora la experiencia del usuario por su flexibilidad para la construcción de consultas [12].
3. Evita que el usuario deba aprender lenguajes de consulta formales [12].

Por otra parte las principales dificultades de este enfoque son:

1. Requiere trasladar el conocimiento del dominio al sistema que lo soporte [12].
2. Debe tratar con ambigüedades e imprecisiones para determinar correctamente los resultados [2], [12], [7].

Algunos ejemplos de este enfoque se describen a continuación:

En [2] se presenta una interfaz de lenguaje natural para bases de datos relacionales. Esta interfaz está compuesta principalmente por los siguientes elementos:

- *Agente conversacional*: su propósito es guiar al usuario en la construcción de las consultas y ayuda identificar y corregir ambigüedades en las mismas. Este agente interactúa con el usuario usando lenguajes natural

(inglés). Esta interacción se realiza usando técnicas de reconocimiento de patrones.

- *Extractor de información*: encargado de mapear el lenguaje natural a consultas para la base de datos. Las consultas son mapeadas a lenguaje soportado por un ORM (*Object Relational Mapping*). Este proceso involucra la limpieza del texto y la división en *tokens* de las consultas en lenguaje natural.
- *Framework ORM*: resuelve la disparidad entre el paradigma orientado a objetos (POO) y las bases de datos relacionales. Mapea las consultas a lenguaje SQL. En este trabajo en particular se usa el ORM Hibernate. Se incluye un framework de este tipo, pues el POO ofrece un modelo lógico más cercano a los usuarios.

Ejemplos de una consulta aceptada por este lenguaje es:

1 List publications of professor Keely

En este trabajo se describen los desafíos que puede representar el tratamiento de consultas escritas en lenguaje natural, tales como la ambigüedad. Con respecto a la evaluación de la propuesta, se evaluó la correctitud de los resultados de un conjunto de consultas y algunos aspectos de usabilidad. En la evaluación participaron 15 personas. A pesar de que se reportan resultados positivos no se compara con otros lenguajes similares o incluso con SQL. Teniendo en cuenta que este trabajo añade dos capas de abstracción, puesto que requiere un proceso de conversión de lenguaje natural a lenguaje de ORM y luego de este a SQL, es de suponer un aumento en tiempo de procesamiento. Sin embargo, en el trabajo no se describe este aspecto.

En [12] se presenta un sistema inteligente de consulta y minería de bases de datos usando procesamiento de lenguaje natural. Su propósito es evitar que el usuario deba aprender lenguajes como SQL. En este artículo se presenta la arquitectura del sistema propuesto y se describen las etapas de procesamiento de las consultas, para llevarlas de lenguaje natural a SQL. Este proceso involucra, entre otros aspectos, un mapa semántico de la base de datos, identificadores de relación, tratabilidad semántica y el divisor de *tokens*. Un ejemplo de una consulta se presenta a continuación:

1 What employees work in computer department?

En este trabajo se describen algunas dificultades de este procesamiento tales como: ambigüedad en los nombres de las relaciones y los atributos, *tokens* intratables como los relacionados con comparación (estos se tratan y se transforman en operadores o valores fijos, por ejemplo, >, =, <=, 10 : 00) y la necesidad de transmitir un alto nivel de conocimiento del dominio al sistema.

Los autores afirman que el sistema es altamente eficiente produciendo consultas SQL, sin embargo no presentan evidencia que soporte dicha afirmación. Además, no hacen referencia al poder de expresividad del mismo y tampoco presentan pruebas con usuarios finales que evalúen su usabilidad.

Por último, en [7] se propone a ROBOT, un sistema de consulta usando lenguaje natural que se describe una alternativa de consulta de bases de datos orientada al usuario. Este

sistema requiere la intervención de un experto del dominio de aplicación deseado, para construir un diccionario del negocio. Este diccionario soporta el proceso de convertir las consultas imprecisas de los usuarios en consultas formales. De esta manera no sólo se resuelve la disparidad entre el modelo mental de los usuarios y el esquema lógico de la base de datos, sino que se ofrece un sistema de consulta que responde a las necesidades propias de cada dominio de aplicación. Un ejemplo de una consulta se presenta a continuación:

1 Who owns 1971 Fords in New Hampshire?

El proceso de evaluación de este sistema comprendió dos etapas de pruebas, una primera de exploración y la segunda en un ambiente completamente real. Sus resultados fueron bastante positivos, pues este logró resolver adecuadamente el 90 % de la consultas en la segunda etapa. Vale la pena destacar, que en la segunda etapa quienes hacían las consultas eran usuarios reales sin conocimiento en bases de datos. La principal debilidad de este sistema es la necesidad de que una persona del dominio intervenga en el proceso de construcción del diccionario del negocio, pues sus resultados dependerán de el grado de experticia del mismo.

II-C. Conceptual

Este enfoque permite recuperar datos de una base de datos usando consultas que no requieren precisión, los resultados de la consulta son aquellos en son conceptualmente similares a los ^{en}condiciones suministradas por el usuario en dicha consulta. Su principal ventaja, similar al enfoque de procesamiento de lenguaje natural es la posibilidad de escribir consultas sin conocer con exactitud la información de las relaciones de la base de datos.

En [13] el autor presenta un lenguaje declarativo conceptual que no requiere el conocimiento exacto de los nombres de las relaciones de la base de datos. En el artículo se define de forma exhaustiva el lenguaje con soporte en usando el que los autores llaman *SCQL-extended predicate calculus*. Además, esta definición también se soporta con ejemplos de consultas equivalentes en lenguaje SQL.

Las operaciones de las que dispone el lenguaje son composición, unión, diferencia, transformación, selección lógica y selección de asociación. Como en otros, casos las consultas son transformadas a consultas equivalentes en SQL para su procesamiento final. Un ejemplo de una consulta se presenta a continuación:

1 Select all tasks assigned to persons participating in the project MES's team

Con respecto a la evaluación, los autores no presentan pruebas de usabilidad o experiencia de usuario. Tampoco hacen mención sobre el poder de expresividad del lenguaje propuesto.

En [6] se presenta un lenguaje de consulta conceptual desarrollado usando una metodología centrada en el usuario. Para realizar una consulta un usuario solo debe suministrar sólo el conjunto de entidades (relaciones y atributos) y condiciones directamente involucradas en los resultados esperados de la

consulta, es decir que el usuario no debe especificar todas las tablas participantes de la consulta final ni las rutas entre ellas. De esta forma se traslada la carga cognitiva del usuario al gestor de bases de datos. Un ejemplo de una consulta se presenta a continuación:

1 Where supplier supplies parts

El uso de una metodología centrada en el usuario busca desarrollar un lenguaje que considere las necesidades de los usuarios de hoy en día. Se parte del supuesto de que el involucramiento y participación de los usuarios podría resultar un una disposición positiva de los usuarios finales con respecto al sistema, generando confianza en estos para usarlo. Su proceso de desarrollo involucró la construcción de dos prototipos y una versión completa del sistema que soporta el lenguaje.

Una prueba con el primer prototipo permitió, entre otras cosas, identificar términos técnicos alejados que implicaban dificultad en los usuarios para su entendimiento. Algunos ejemplos de estos términos son: fuente o destino. Por su parte, en el segundo prototipo, los términos técnicos fueron reemplazados por unos más comunes o frases que facilitarían su comprensión. Una prueba con este segundo prototipo, permitió concluir que si bien los términos técnicos inicialmente son de difícil comprensión; estos no debe ser quitados de la interfaz, sino acompañados de explicaciones o aclaraciones. En ambas ocasiones se observó un mejor desempeño de los usuarios usando el lenguaje de consulta conceptual, en comparación con SQL. El desempeño fue medido en términos de la precisión de las consultas escritas, el tiempo empleado para escribir las consultas y la percepción del usuario con respecto a la facilidad de uso del lenguaje en cuestión.

La versión final del sistema incluye toda la retroalimentación percibida con los dos prototipos, sin embargo no se incluye evidencia de una evaluación de este sistema.

II-D. Variante de SQL

A diferencia de todos los enfoques anteriores, que añaden una capa adicional que busca esconder el lenguaje SQL con un lenguaje diferente o una interfaz al mismo, este enfoque extiende a SQL para permitir la introducción de nombres de relaciones y atributos inexactos; manteniendo una sintaxis similar a la del lenguaje SQL, siendo esta su principal ventaja.

En [4] los autores presentan un lenguaje de consulta que denominan SQL libre de esquema (*Schema-free SQL*). Este lenguaje permite expresar consultas usando un conocimiento parcial del esquema lógico de la base de datos y de las rutas de *join* necesarias para obtener los resultados esperados. Para lograr este propósito los autores proponen seguir la sintaxis del lenguaje SQL con dos relajaciones:

- *Relajación del esquema*: los usuarios no tiene que especificar los nombres exactos de los elementos de la consulta, tales como nombres de relaciones y de atributos. Para esto se puede acompañar los nombres candidatos acompañados de un signo de interrogación ?. Si el usuario no tiene idea de un nombre puede usar *x?* o simplemente ?.

- *Relajación de la ruta de join*: los usuarios no deben especificar cuáles relaciones están involucradas, ni cómo ellas se debe unir entre sí.

Un ejemplo de una consulta se presenta a continuación:

```

1 SELECT count( actor?.name?)
2 WHERE actor?.gender?="male"
3     and director?.name?="James Cameron"
4     and produce_company?="20th Century Fox"
5     and year? >1995
6     and year? < 2005

```

El sistema que soporta el lenguaje debe mapear nombres de relaciones inexactos a sus nombres exactos. Para este propósito se usan funciones de similitud. Una vez se cuenta con los nombres mapeados se construye una red de posibles rutas de *joins*, dado que el conjunto de posibles rutas puede ser muy grande, se selecciona solo el *top k* de estos usando una estimación de el potencial que puedan tener las rutas. Esta estimación se basa pesos asignados a las rutas mientras se construye la red de *joins*.

La ventaja de este lenguaje es que aprovecha la expresividad y poder de SQL, sin que se requiera tener un conocimiento exacto de su esquema lógico. Aliviando así, la complejidad derivada de procesos de normalización, que aunque suponen un buen diseño de base de datos, alejan los esquemas lógicos de las bases de datos, de los modelos mentales de los usuarios.

La evaluación de este lenguaje se basa en dos aspectos: la efectividad en términos de las consultas SQL generadas correctamente y la carga cognitiva de los usuarios representada en el número de elementos (por ejemplo: nombres de relaciones o de atributos) usados en la consulta. Los resultados muestran que usando un $k=10$, el 100 % de las consultas generadas fueron correctas. Y el número de elementos usados en la consulta siempre fue menor para el SQL libre de esquema que para el SQL.

II-E. Basado en palabras clave

Este enfoque es quizá el más conocido y accedido por usuarios novatos. Debido a que permite recuperar datos de forma fácil e intuitiva a los usuarios. Los usuarios se evitan la necesidad de escribir consultas complejas, pues solo deben suministrar palabras clave relacionadas con el resultado esperado [4], [3]. Un ejemplo de estos son los buscadores que se encuentran en los sitios web. Este enfoque tiene entonces una buena usabilidad, sin embargo, no resuelve todos los desafíos que representa la consulta de bases de datos, debido a que el uso de booleanos o frases en comillas no son suficientes para cuando se requiere información muy precisa [3]. En este enfoque el usuario es tolerante a resultados irrelevantes y tampoco se preocupan si algún resultado relevante falta, siempre y cuando algunos de los resultados si sean relevantes [3]. Sin embargo, esta tolerancia no es aceptable en todos los casos.

Las principales ventajas de este enfoque son entonces, su simplicidad y que los usuarios no requieren conocer el esquema lógico de la base de datos para escribir una consulta.

Este enfoque ha sido incluido con el objetivo de resaltar la importancia de que los medios de consulta de una base

de datos se encarguen tanto de la usabilidad de los mismos, como de resolver necesidades precisas de recuperación de información.

III. ANÁLISIS

En la sección anterior se presentaron cinco enfoques que buscan resolver la disparidad entre el modelo mental de los usuarios y los esquemas lógicos de las bases de datos. Para analizar los diferentes enfoques, es posible basarse en los requerimientos derivados del acceso a la información por parte de usuarios novatos. A continuación se presenta dicho análisis:

- *Recuperar información en un tiempo corto* [2]: en cuanto a este requerimiento es difícil realizar un análisis objetivo solamente uno de los trabajos presentados reportó resultados de rendimiento [9]. sin embargo en el mismo artículo es superado por otros trabajos. Además no se compara sistemas que soporte lenguajes de uso extendido como SQL. De este y los demás trabajos es posible suponer que rendimiento es inferior al de sistemas que soportan lenguajes como SQL, pues en su mayoría añaden una o varias capas de abstracción y traducen las consultas a lenguajes como SQL. Adicionalmente, ninguno de los trabajo menciona detalle alguno respecto a la optimización de consultas, algunos trabajos describen estrategias orientadas a disminuir los tiempos de procesamiento en la traducción de las consultas a lenguajes de más bajo nivel como SQL. Sería importante determinar el impacto en términos de tiempos de respuestas de los diferentes trabajos, para lograr un análisis más objetivo.
- *Un medio de consulta fácil de usar y entender* [2]: en este aspecto los trabajos ofrecen información relacionada con facilidad de uso y algunos de estos incluso reportan resultados relacionados con la carga cognitiva del usuario para formular consultas. Aquellos trabajos que realizaron pruebas en las que compararon el trabajo propuesto y lenguajes de más bajo nivel como SQL, reportaron resultados positivos para sus trabajos. La posibilidad de escribir consultas teniendo conocimiento impreciso del esquema lógico de la base de datos, disminuye la carga cognitiva de los usuarios y los tiempos de formulación de consultas, pues los usuarios pueden usar términos de negocio que comúnmente usan, o pueden escribir las consultas en su propio lenguaje o diligenciar formularios, presionar botones o interactuar con grafos. Sin embargo, es difícil decidir cuál enfoque podría resultar mejor que otro. Una razón para esto es que cada trabajo se evaluó de forma particular, usando métodos diferentes. Además, algunos incluyeron participantes que representaban usuarios finales, otros incluyeron participantes provenientes del área de las tecnologías de la información y otros trabajos fueron evaluados por sus autores. Además, el trabajo con más usuarios involucrados (de acuerdo con lo reportado), involucró a 38 participantes. Finalmente, algunos trabajos no reportaron evaluaciones con respecto a este requerimiento. En consecuencia, no es posible

obtener conclusiones sólidas respecto a la facilidad de uso y de aprendizaje de estos enfoques.

- *Recuperar resultados precisos y completos [3]:* Con respecto a este solo uno de los trabajos reporto un grado de efectividad del 100 % [4]. Este trabajo la efectividad se evaluó determinando si dada una consulta de entrada el sistema era capaz de construir la consulta SQL esperada por los autores. Otros trabajos reportaron resultados positivos, pero con menor grado de efectividad. Otro aspecto que permite analizar este requerimiento es el poder de expresividad de los trabajos propuestos. Al respecto, solo dos trabajos reportan grados de expresividad comparable al lenguaje SQL. Los demás trabajos en su mayoría permiten consultar las bases datos involucrando varias relaciones, atributos y suministrando condiciones de filtrado, pero no reportan grados de expresividad o no son precisos al respecto. Adicionalmente, estos trabajos no suministran información sobre funciones agregadas como las disponibles en SQL.
- *Representación multidimensional de los resultados, tales como tablas o grafos [3]:* Con respecto a este aspecto en los sistemas relacionados en el enfoque visual son precisos respecto a la presentación de los resultados que pueden producir representaciones en forma de tablas o grafos. En los demás enfoques no se precisa la representación de los resultados, pues este depende de los sistemas que soporten o usen los lenguajes de consulta.

Lo anterior indica que los trabajos presentados no cuentan con evidencia suficiente que permitan concluir que los enfoques descritos mejoran de forma contundente la usabilidad de los sistemas de consulta de bases de datos o la experiencia de los usuarios.

Finalmente, es conveniente resaltar que ninguno de los enfoques y sus respectivos trabajos ha sido ampliamente usado en ambientes reales. Muchos de estos son sólo prototipos que carecen de la robustez de sistemas como los gestores de bases de datos comerciales. Esto puede deberse a dos razones expuestas por [15]. Primero, cuando las propuestas nuevas no ofrecen mejoran el rendimiento de forma significativa o no ofrecen ventajas de funcionalidad, es muy probable que no sean acogidas. Segundo, los debates técnicos generalmente son resueltos por los "grandes" del mercado.

IV. CONCLUSIONES

En este artículo se presentó una clasificación de enfoques de consulta de bases de datos. Estos enfoques están orientados a permitir que usuarios novatos también puedan consultar las bases de datos. La principal estrategia para lograr esto es adicionar modelos o capas de más alto nivel, que permitan resolver la disparidad entre el esquema lógico de las bases de datos y los modelos mentales de los usuarios, mejorando la usabilidad de los sistemas de consulta y la experiencia de usuarios. Para cada enfoque se presentaron algunos trabajos relacionados, sus propuestas, ventajas y desventajas.

En general estos trabajos presentan resultados positivos en términos de su facilidad de uso y efectividad para recuperar

los resultados esperados por los usuarios. Sin embargo, estos resultados no son lo suficientemente significativos y para concluir de forma sólido sobre sus ventajas y posibilidades de ser usados a la escala de lenguajes de consulta como SQL. Ninguno de los trabajos presenta evidencias significativas que indiquen estos enfoques representan una mejoría contundente en términos de usabilidad y experiencia de usuario.

Finalmente, con este trabajo se pudo evidenciar la importancia de que cualquier propuesta novedosa en cualquiera de estos enfoques, debe preocuparse no sólo por ofrecer un medio de consulta amigable a los usuarios, sino que también debe representar ventajas significativas en términos de rendimiento o de funcionalidad. De lo contrario, los grandes del mercado seguirán determinando los lenguajes y sistemas usados ampliamente.

REFERENCIAS

- [1] J. Lloret-Gazo, "A Survey on Visual Query Systems in the Web Era," 2016, pp. 343–351. [Online]. Available: http://link.springer.com/10.1007/978-3-319-44406-2_{_}28
- [2] A. Alghamdi, M. Owda, and K. Crockett, "Natural Language Interface to Relational Database (NLI-RDB) Through Object Relational Mapping (ORM)," 2017, pp. 449–464. [Online]. Available: http://link.springer.com/10.1007/978-3-319-46562-3_{_}29
- [3] H. V. Jagadish, A. Chapman, A. Elkiss, M. Jayapandian, Y. Li, A. Nandi, and C. Yu, "Making database systems usable," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data - SIGMOD '07*. New York, New York, USA: ACM Press, 2007, p. 13. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1247480.1247483>
- [4] F. Li, T. Pan, and H. V. Jagadish, "Schema-free SQL," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data - SIGMOD '14*. New York, New York, USA: ACM Press, 2014, pp. 1051–1062. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2588555.2588571>
- [5] T. CATARCI, M. F. COSTABILE, S. LEVIALDI, and C. BATINI, "Visual Query Systems for Databases: A Survey," *Journal of Visual Languages & Computing*, vol. 8, no. 2, pp. 215–260, apr 1997. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1045926X97900379>
- [6] V. Owei, "Development of a Conceptual Query Language: Adopting the UserCentered Methodology," *The Computer Journal*, vol. 46, no. 6, pp. 602–624, jun 2003. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.105.2806>
- [7] L. Harris, "User oriented data base query with the ROBOT natural language query system," *International Journal of Man-Machine Studies*, vol. 9, no. 6, pp. 697–713, nov 1977. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0020737377800370>
- [8] M. C. Pabón and C. A. Collazos, "A Visual Query Language for Data Graphs," in *Proceedings of the XVI International Conference on Human Computer Interaction - Interacción '15*, vol. 07-09-Sept. New York, New York, USA: ACM Press, sep 2015, pp. 1–2. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2829875.2829918>
- [9] H. Störrle, "VMQL: A visual language for ad-hoc model querying," *Journal of Visual Languages & Computing*, vol. 22, no. 1, pp. 3–29, feb 2011. [Online]. Available: <http://www.sciencedirect.com.bd.univale.edu.co/science/article/pii/S1045926X1000073X>
- [10] S. Keng, "A visual object-relationship query language for user-database interaction," *Telematics and Informatics*, vol. 15, no. 1-2, pp. 103–119, feb 1998. [Online]. Available: <http://www.sciencedirect.com.bd.univale.edu.co/science/article/pii/S0736585398000070>
- [11] C. R. Borges and J. A. Macías, "Feasible database querying using a visual end-user approach," in *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems - EICS '10*. New York, New York, USA: ACM Press, 2010, p. 187. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1822018.1822047>

- [12] R. Agrawal, A. Chakkarwar, P. Choudhary, U. A. Jogalekar, and D. H. Kulkarni, "DBIQS - An intelligent system for querying and mining databases using NLP," in *2014 International Conference on Information Systems and Computer Networks (ISCON)*. IEEE, mar 2014, pp. 39–44. [Online]. Available: <http://ieeexplore.ieee.org/document/6965215/>
- [13] V. Ovchinnikov, "A Concept-Based Query Language Not Using Proper Association Names," 2006, pp. 374–400. [Online]. Available: <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59140-935-9.ch019>
- [14] B. Ding, B. Zhao, C. X. Lin, J. Han, C. Zhai, A. Srivastava, and N. C. Oza, "Efficient Keyword-Based Search for Top-K Cells in Text Cube," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 12, pp. 1795–1810, dec 2011. [Online]. Available: <http://ieeexplore.ieee.org/document/5710919/>
- [15] M. Stonebraker and J. Hellerstein, "What goes around comes around," *Readings in Database Systems*, vol. 4, 2005.