SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Ciudad Madero

INSTITUTO TECNOLÓGICO DE CIUDAD MADERO
División de Estudios de Posgrado e Investigación

"MÓDULO PARA COMPOSICIÓN DE CONSULTAS PARA UNA INTERFAZ DE LENGUAJE NATURAL A BASES DE DATOS"

OPCIÓN I
Tesis Profesional

Que para obtener el grado de
Maestro en Ciencias de la Computación

Presenta
Lic. Alan Gabriel Aguirre Lam

Director de Tesis:
Dr. Rodolfo Abraham Pazos Rangel

Cd. Madero, Tamaulipas, México Marzo 2015

"2015, Año del Generalísimo José María Morelos y Pavón"

Cd. Madero, Tamps; a 27 de Febrero de 2015.

OFICIO No.: U5.045/14
AREA: DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS

LIC. ALAN GABRIEL AGUIRRE LAM
NO. DE CONTROL G06071536
P R E S E N T E

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su examen de grado de Maestría en Ciencias de la Computación, el cual está integrado por los siguientes catedráticos:

PRESIDENTE :	DRA.	GUADALUPE CASTILLA VALDEZ
SECRETARIO :	DR.	JUAN JAVIER GONZÁLEZ BARBOSA
VOCAL :	DR.	RODOLFO ABRAHAM PAZOS RANGEL
SUPLENTE	DR.	JOSÉ ANTONIO MARTÍNEZ FLORES
DIRECTOR DE TESIS :	DR.	RODOLFO ABRAHAM PAZOS RANGEL

Se acordó autorizar la impresión de su tesis titulada:

"MÓDULO PARA COMPOSICIÓN DE CONSULTAS PARA UNA INTERFAZ DE LENGUAJE NATURAL A BASES DE DATOS"

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con Usted el logro de esta meta.

Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

A T E N T A M E N T E
"POR MI PATRIA Y POR MI BIEN"®

Ma. Yolanda Chávez Cinco

M. P. MARÍA YOLANDA CHÁVEZ CINCO
JEFA DE LA DIVISIÓN

S.E.P.
DIVISION DE ESTUDIOS
DE POSGRADO E
INVESTIGACION
I T C M

c.c.p.- Archivo
Minuta
MYCHC 'NLCO ' jan

Contenido

1. Introducción 1
1.1 Antecedentes 2
1.2 Planteamiento del problema 4
1.3 Objetivos 5
1.4 Justificación y beneficios 6
1.5 Alcance y limitaciones 6

2. Marco conceptual y estado del arte 8
2.1 Marco conceptual 8
2.1.1 Bases de datos 9
2.1.2 Lenguaje de consultas estructurado 9
2.1.3 Lenguaje de definición de datos 9
2.1.4 Lenguaje de manipulación de datos 10
2.1.3 ILNBDs 10
2.1.4 Composición y formulación de consultas 11
2.1.5 Diálogo 11
2.2 Trabajos relacionados 12
2.2.1 CoBase - UCLA (1999) 12
2.2.2 Query Builder - University of Calgary (2004) 13
2.2.3 WYSIWYM - The Open University (2007) 14
2.2.4 TAICHI – IBM T.J. Watson Research Center (2010) 15
2.2.5 Microsoft Access QBE (2013) 15
2.2.6 Conclusiones 16

3. Metodología de solución 18
3.1 Arquitectura propuesta 18
3.2 Clasificación de las tablas de una base de datos 19
3.3 Tipos de consultas a tratar 20
3.4 Árbol de composición 20
3.4.1 Construcción 21
3.4.2 Características 22

4. Interfaz de composición 24
4.1 Conceptos para la composición de consultas 24
4.2 Selección del tema de interés 25
4.3 Selección de elementos de interés 26
4.4 Definición de condiciones de búsqueda 29
4.5 Generación de la consulta en SQL 32
4.6 Vista previa y resultados de la consulta 34
4.7 Proceso de composición de consultas 35

5. Interfaz de configuración 38
5.1 Configuración de tablas 39
5.2 Configuración de columnas 40
5.3 Tipos de dato personalizados 41
6. Experimentación 44

6.1 Pruebas sobre consultas del tipo 1 al 4 44
6.1.1 Objetivo de las pruebas 44
6.1.2 Descripción del ambiente de pruebas 44
6.1.3 Resultados 46
6.1.4 Conclusiones 48
6.2 Pruebas funcionales sobre consultas del tipo 5 48
6.2.1 Descripción de las pruebas 48

7. Conclusiones y trabajos futuros 51
7.1 Conclusiones 51
7.2 Trabajos futuros 52

APÉNDICES

Apéndice A. Descripción de la base de datos ATIS 54
Apéndice B. Corpus de consultas de la base de datos ATIS 61
Apéndice C. Descripción del diccionario de información semántica 63

Referencias 65

Lista de Tablas

Tabla 2.1. Trabajos relacionados con la composición de consultas 17
Tabla 3.1. Clasificación de tablas 20
Tabla 3.2. Tipos de consultas 20
Tabla 4.1. Conceptos involucrados en la composición de consultas 24
Tabla 4.2. Componentes de la ventana Selección del tema de interés 25
Tabla 4.3. Clasificación de tablas para la base de datos ATIS 26
Tabla 4.4. Componentes de la ventana Selección de elementos de interés 26
Tabla 4.5. Componentes de la ventana Definición de condiciones de búsqueda 29
Tabla 5.1. Información modificable mediante la interfaz de configuración 38
Tabla 6.1. Resultados por consulta 46

Lista de Figuras

Figura 1.1. Arquitectura general de la ILNBD desarrollada por Aguirre 2
Figura 2.1. Flujo en una ILNBD 11
Figura 2.2. Grafo semántico de una BD de transporte 12
Figura 2.3. Consulta formulada en la interfaz de CoBase 13
Figura 2.4. Interfaz principal de Query Builder 14
Figura 2.5. Editor de consultas de WYSIWYM 14
Figura 2.6. Ejemplo de una consulta visual en TAICHI 15
Figura 2.7. Interfaz QBE de MS Access 16
Figura 2.8. Esquema de la base de datos ATIS 17
Figura 3.1. Arquitectura propuesta de la interfaz de composición 19
Figura 3.2. Ejemplo de un árbol de composición 22
Figura 4.1. Componentes de la ventana Selección del tema de interés 25
Figura 4.2. Componentes de la ventana Selección de elementos de interés 27
Figura 4.3. Construcción de la ruta de un nodo seleccionado 27
Figura 4.4. Ejemplo de la construcción de la ruta de un elemento de interés 28
Figura 4.5. Vector de elementos de interés 28
Figura 4.6. Componentes de la ventana Definición de condiciones de búsqueda 30
Figura 4.7. Llenado de la lista de operadores de comparación 30
Figura 4.8. Definición de una condición de búsqueda 31
Figura 4.9. Lista de condiciones de búsqueda 32
Figura 4.10. Construcción de la cláusula Select. 33
Figura 4.11. Construcción de restricciones en la cláusula Where 33
Figura 4.11. Construcción de las reuniones entre tablas en la cláusula Where 34
Figura 4.12. Construcción de la cláusula From 34
Figura 4.13. Vista previa y resultado de la consulta 35
Figura 4.14. Selección del tema de interés 36
Figura 4.15. Selección de los elementos de los temas 36
Figura 4.16. Especificación de las condiciones de búsqueda. 37
Figura 5.1. Interfaz de configuración (tablas) 39
Figura 5.2. Ejemplo de configuración de una tabla 39
Figura 5.3. Interfaz de configuración (columnas) 40
Figura 5.4. Ejemplo de configuración de una columna 41

Figura 5.5. Interfaz de configuración (tipos de dato)..... 42
Figura 5.6. Ejemplo de creación de un tipo de dato..... 42
Figura 6.1. Funcionamiento de la interfaz para pruebas..... 45
Figura 6.2. Promedio de intentos y tiempo por consulta..... 47
Figura A.1. Esquema de la base de datos ATIS..... 61
Figura C.1. Esquema del diccionario de información semántica..... 64

Capítulo 1

Introducción

A medida que pasa el tiempo, el uso de las computadoras se va incrementando con la introducción de nuevos dispositivos y tecnología desarrollada constantemente. Actualmente la mayoría de los hogares y empresas cuentan con equipos de cómputo para satisfacer sus necesidades de información. Muchas de estas necesidades están relacionadas con consultas a bases de datos (BDs), las cuales requieren que el usuario tenga una capacitación para poder hacer uso de ellas; sin embargo, no todos los usuarios pueden tener acceso a este tipo de capacitación.

Para facilitar el uso de las BDs, se han desarrollado distintas herramientas que pueden ser usadas por usuarios inexpertos sin necesidad de recibir capacitación sobre lenguajes de consulta a bases de datos, p. ej., SQL (por sus siglas en inglés *Structured Query Language*). Algunas herramientas permiten hacer uso del lenguaje natural (LN) como medio para consultar bases de datos. Este tipo de software es llamado interfaces de lenguaje natural a bases de datos (ILNBD). Sin embargo, el desarrollo de este tipo de interfaces no ha sido el deseado, debido a lo complejo que resulta procesar el contenido semántico del lenguaje natural y utilizarlo en una ILNBD [Pazos, 2014].

Por lo expuesto anteriormente, se pueden señalar algunos problemas que usualmente tienen los usuarios al realizar consultas por medio de lenguaje natural; por ejemplo, el usuario puede desconocer la cobertura de lenguaje natural que tiene la ILNBD y formular consultas correctamente pero la interfaz sería incapaz de comprender el significado de la consulta, debido a que una ILNBD sólo funciona con un subconjunto del LN y el usuario estaría planteando consultas que no se encuentran dentro de dicho subconjunto. Otra situación posible es que el usuario formule consultas cuyo resultado esperado no se encuentra dentro de la BD. Otro ejemplo más podría ser que la ILNBD interprete de manera errónea una consulta planteada por el usuario debido a la forma en que se encuentra redactada la consulta.

El objetivo de este trabajo se centra en el desarrollo de una herramienta intuitiva y fácil de usar que ayude al usuario inexperto a componer consultas en SQL correctamente sin necesidad de tener conocimientos sobre el mismo ni del contenido de la BD.

1.1 Antecedentes

Esta tesis forma parte de un proyecto denominado Interfaz de Lenguaje Natural Español hacia Bases de Datos para Usuarios de Internet, el cual comenzó a desarrollarse en el CENIDET en septiembre del 2001, y se continúa desarrollando en el ITCM desde agosto del 2002. Dicho proyecto pretende implementar una interfaz que permita a usuarios casuales e inexpertos consultar BDs mediante expresiones en lenguaje natural en español no acotado.

La interfaz desarrollada en el ITCM por González en el 2005 consiste en utilizar un módulo de traducción para mejorar el porcentaje de consultas contestadas correctamente y permitir que la ILNBD sea independiente de la BD [González, 2005]. Lo anterior se logra mediante un módulo de traducción y un preprocesador que genera automáticamente un diccionario de dominio. Sin embargo, esta interfaz no cuenta con un diálogo de aclaración en caso de que el usuario introduzca una consulta que no puede ser entendida por la interfaz, lo cual limita su capacidad para obtener un mejor porcentaje de aciertos.

En el año 2006 Rojas desarrolló una mejora en la interfaz desarrollada por González [Rojas, 2006], que incorpora procesos de diálogo independientes de dominio para un administrador de diálogo para la ILNBD. Dichos diálogos permiten resolver la elipsis semántica en las consultas formuladas por el usuario mediante ciclos de pregunta-respuesta para obtener los resultados esperados.

Posteriormente en 2014 Aguirre propuso una arquitectura para el diccionario de información semántica (estructura utilizada para almacenar la información semántica de la BD) basada en un nuevo modelo semánticamente enriquecido de bases de datos [Aguirre, 2014], el cual incluye información semántica relevante para la traducción de consultas. Además, esta versión cuenta con una arquitectura basada en capas de funcionalidad (similares a las del modelo OSI para redes de comunicación para computadoras), donde cada capa funcional se ocupa de un problema en el proceso de traducción. Las capas funcionales de la arquitectura incluyen la mayoría de los problemas encontrados en el proceso de traducción de lenguaje natural a SQL.

En la figura 1.1 se muestra la arquitectura de la interfaz desarrollada por Aguirre. Es conveniente aclarar que en el Módulo de Traducción en Capas de Funcionalidad existen elementos en desarrollo.

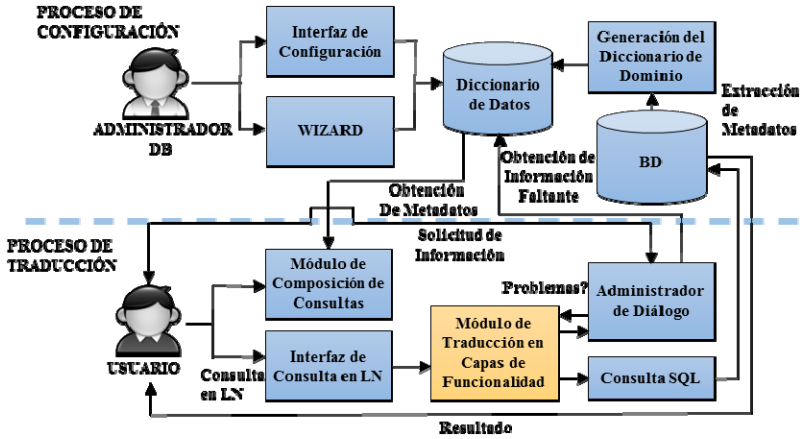


Figura 1.1. Arquitectura general de la ILNBD desarrollada por Aguirre

En la arquitectura presentada en la Figura 1.1 se pueden diferenciar dos procesos principales denominados proceso de configuración y proceso de traducción. En primer lugar es necesario realizar una configuración previa al uso de la ILNBD por los usuarios, debido que ésta es independiente de dominio y debe ser configurada correctamente para que las consultas puedan ser interpretadas correctamente. El proceso de configuración es efectuado por el administrador de BDs, quien a través de la interfaz de configuración debe seleccionar la BD sobre la que trabajará la ILNBD. Posteriormente, si es la primera vez que la ILNBD utiliza la BD seleccionada, el administrador de BDs debe solicitar a la ILNBD la generación del dominio (diccionario de datos).

Una vez generado el dominio de la BD el administrador puede interactuar directamente con éste para configurarlo, o en su defecto puede usar el *Wizard*, en el cual se toma una consulta en LN y la interfaz solicita se introduzca la consulta correcta en SQL para hacer un proceso de configuración automático basado en la expresión en SQL proporcionada por el administrador.

Posteriormente, con la ILNBD configurada correctamente, el usuario final puede hacer uso de ésta formulando una consulta en LN por medio de la interfaz de consulta en LN. La consulta formulada por el usuario es introducida al módulo de traducción, el cual procesa la consulta mediante las capas de funcionalidad.

Ocasionalmente en el módulo de traducción se detectan problemas relacionados con la consulta, los cuales se pasan al administrador de diálogo para solicitar una aclaración por parte del usuario en donde especifique algunos aspectos de la consulta.

Al terminar el procesamiento de la consulta, si la consulta ha sido procesada correctamente, el módulo de traducción retorna como resultado una consulta en SQL, la cual se pasa al sistema manejador de bases de datos (SMBD) para obtener los resultados de dicha consulta. Una vez obtenidos los resultados de la consulta, éstos son presentados al usuario.

Además de las ILNBDs, existen otras herramientas llamadas Interfaces QBE (Query by Example, por sus siglas en inglés), las cuales tienen el objetivo de facilitar al usuario la formulación de consultas a BDs con conocimiento básico de la BD. En este tipo de interfaces, el método para definir consultas a BDs es más visual, haciendo uso de una cuadrícula a forma de ejemplo, la cual asemeja una expresión en SQL; la cuadrícula es llenada por el usuario para obtener una expresión en SQL, la cual es procesada por un SMBD.

En el año 1999, se desarrolló una interfaz QBE [Rasgado, 1999], en la que se propone una herramienta para usuarios inexpertos, orientada a la formulación de consultas a BDs en Internet. Posteriormente, en el 2000 se propuso una mejora en la interfaz [May, 2000], la cual consiste en mejorar el diseño de ésta para que fuera más amigable y además permitiera la formulación de una consulta que involucre tablas de dos BDs. En esta interfaz es posible formular y ejecutar consultas a BDs y para ello se pueden distinguir tres fases:

- Selección de las BDs y tablas participantes en la consulta.
- Formulación de la consulta.
- Ejecución de la consulta y presentación de resultados.

En la Figura 1.2 se puede apreciar la interfaz de usuario para formular consultas basadas en ejemplos, la cual muestra en la parte inferior la formulación de una consulta y en la parte superior las tablas involucradas en la formulación de la misma.

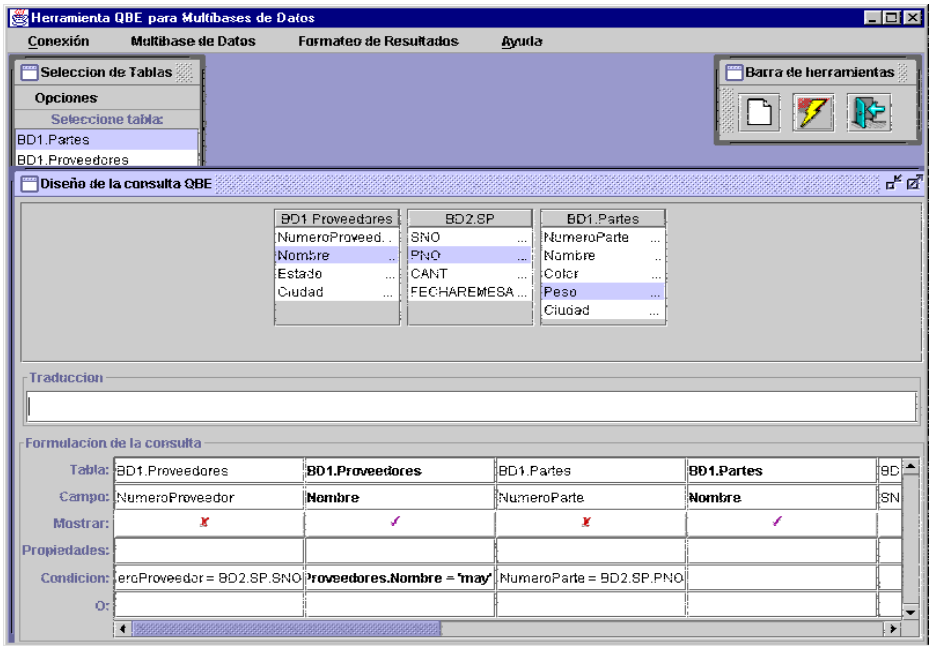


Figura 1.2. Interfaz de formulación de consultas basadas en ejemplos.

El proyecto de esta tesis está basado en la composición de una consulta en SQL con alguna semejanza a los QBE, integrando la composición de consultas a una ILNBD. Esto permite al usuario componer consultas sin los problemas que pueden presentar las ILNBDs.

1.2 Planteamiento del problema

En una ILNBD el usuario hace una solicitud a una BD por medio de lenguaje natural; para tal efecto, el usuario debe tener una consulta previamente formulada. Sin embargo, una ILNBD también está orientada para servir a usuarios casuales e inexpertos, los cuales pueden llegar a tener dificultades para formular consultas, ya sea por desconocer el contenido de la BD, desconocer el esquema de la BD, o simplemente tener dificultad para expresar la consulta en lenguaje natural. También al formular consultas en lenguaje natural, el usuario se puede ver en la necesidad de elegir de qué manera plantear su consulta debido a que las consultas expresadas en lenguaje natural son sensibles a errores de redacción (p. ej., errores de ortografía, errores de gramaticales) o procesamiento (a nivel léxico, sintáctico, o semántico) [Hallet, 2007].

El módulo de composición de consultas desarrollado en esta tesis opera mediante un proceso de diálogo iterativo entre el módulo y el usuario, en el cual, las respuestas introducidas por el usuario se toman en cuenta para construir la consulta en SQL.

Para ejemplificar el problema descrito, se plantea la siguiente consulta sobre la BD Geobase:

¿Qué ciudades atraviesa el río Mississippi?

La BD Geobase cuenta con información sobre la geografía de los Estados Unidos de América. La información contenida en dicha BD es la siguiente:

- Estados.- Sus capitales, poblaciones, áreas, densidades de población, ciudades, ríos y estados colindantes.
- Ciudades.- Sus poblaciones y los estados en que se encuentran.
- Ríos.- Sus longitudes y los estados que son atravesados por éstos.
- Montañas.- Sus alturas y los estados en que se encuentran.

De acuerdo con la información presentada en los puntos anteriores, se puede deducir que de un río, en este caso el "*río Mississipi*" se puede conocer a través de qué estado fluye éste, sin embargo, no existe información que indique sobre qué ciudad o ciudades atraviesa dicho río.

Por lo anterior, el usuario puede pensar que su consulta puede ser fácilmente contestada por la ILNBD; sin embargo, esto no es así debido a que no existe la información solicitada dentro de la BD. Por lo tanto, el problema se encuentra en la consulta planteada por el usuario y no en la capacidad de procesamiento de la ILNBD.

Este tipo de errores son cometidos por los usuarios debido a que desconocen el contenido y el esquema de la BD.

1.3 Objetivos

Para resolver el problema presentado en este trabajo, se plantea el siguiente objetivo general (OG):

OG) Diseñar e implementar una herramienta intuitiva y con independencia de dominio, que facilite a los usuarios inexpertos la formulación de consultas a bases de datos mediante la implementación de un módulo que permita a una ILNBD [Aguirre, 2014] ofrecer el servicio de composición de dichas consultas en SQL.

Para alcanzar el objetivo general (OG), se requiere alcanzar los siguientes objetivos específicos (OEs):

OE.1) La herramienta debe facilitar a los usuarios componer consultas a una base de datos mediante una interfaz fácil de usar.

OE.2) La herramienta desarrollada debe permitir componer consultas de manera no restringida, es decir, sin uso de plantillas ni consultas preestablecidas.

OE.3) El proceso de composición de consultas debe obtener como resultado una consulta escrita en SQL.

OE.4) Las consultas compuestas por la herramienta deben tener coherencia con la estructura de la BD.

1.4 Justificación y beneficios

Gran cantidad de información es utilizada a cada instante para tomar decisiones en la vida diaria (p. ej., empresas financieras, telefónicas, etc.). Mucha de esta información es almacenada en bases de datos, con esto surge la necesidad de crear sistemas de consulta que sean eficaces para obtener la información que solicitan los usuarios. Desafortunadamente, para operar estos sistemas, es necesario proporcionar a los usuarios medios que les faciliten el proceso de consulta, ya que no se puede esperar que aprendan y usen lenguajes formales de consulta, tales como SQL, que suelen ser utilizados por expertos en computación.

Los beneficios principales de una ILNBD son los siguientes:

- Facilitar a los usuarios casuales o inexpertos consultar BDs por medio de lenguaje natural.
- No requieren que el usuario aprenda un lenguaje artificial de comunicación.
- Permiten al usuario formular consultas en SQL cuya estructura es compleja.

Como se puede observar, las ILNBD proveen a los usuarios la facilidad de realizar consultas en lenguaje natural. Sin embargo, en algunos casos los usuarios casuales o inexpertos pueden desconocer cómo se realiza correctamente la formulación de una consulta. Por lo tanto, es necesario guiar a dichos usuarios en la formulación de la misma mediante una interfaz gráfica. En tal tarea se centra el presente trabajo de investigación.

Los beneficios principales del presente trabajo son los siguientes:

- Proporcionar a una ILNBD la capacidad de componer consultas en SQL.
- Permitir a los usuarios inexpertos componer consultas formuladas correctamente a través del módulo de composición de consultas.
- Asegurar que las consultas formuladas en SQL por el módulo de composición de consultas puedan ser respondidas de manera satisfactoria al ser procesadas por un sistema manejador de bases de datos.
- Evitar que los usuarios inexpertos cometan errores relacionados con la entrada de texto libre.

1.5 Alcance y limitaciones

El alcance de este proyecto se define en los siguientes puntos:

a) El módulo para composición de consultas debe funcionar para la ILNBD [Aguirre, 2014], y debe ser independiente de dominio.
b) El módulo para composición de consultas debe proporcionar a la ILNBD [Aguirre, 2014] la capacidad de componer consultas en SQL correctamente.
c) El módulo para composición de consultas debe proporcionar al usuario inexperto la información necesaria sobre la BD para poder componer consultas.

d) El módulo para composición de consultas debe componer solamente consultas coherentes con el contenido de la BD.
e) El módulo para composición de consultas debe funcionar con BDs complejas (p. ej., la base de datos ATIS).

Las limitaciones de este proyecto se definen en los siguientes puntos:

a) El idioma del módulo para composición de consultas es únicamente español.
b) La interfaz no proporciona información que no se encuentre explícitamente en la BD para la composición de consultas; es decir no se tratan consultas de bases de datos deductivas.
c) No se tratan consultas de bases de datos temporales.
d) No se consideran consultas que involucren funciones de agregación y agrupamiento (Group By).
d) No se tratan los tipos de consultas del 6 al 9 mencionados en la subsección 3.3.
e) No se considera el problema de transformar una consulta en SQL a su equivalente optimizada.
f) Sólo se pretende realizar pruebas de funcionalidad al módulo de composición. Debido a que el resultado retornado por el módulo es una consulta en SQL, y una consulta compuesta por el usuario es construida a partir de conceptos; por lo tanto resulta difícil evaluar el resultado de dicha consulta.

Capítulo 2

Marco conceptual y estado del arte

Tal como se mencionó en el Capítulo 1, las ILNBDs han sido herramientas útiles para la obtención de información de BDs. Al igual que las ILNBDs descritas anteriormente, existen otras herramientas que permiten obtener información a partir de una BD. Estas herramientas hacen uso de distintas técnicas para obtener una consulta en algún lenguaje de consulta formal (p. ej., SQL) y reciben el nombre de constructores visuales de consultas (VQB por sus siglas en inglés).

Los VQB hacen uso de interfaces gráficas de usuario amigables para la composición de consultas. Dichas interfaces de usuario tienen diversas características y técnicas de composición, entre ellas destacan las siguientes:

- Diseño *drag and drop* de los elementos del esquema de la BD.
- Uso de plantillas prestablecidas en LN para componer parte de la consulta.
- Asociación de funciones en SQL con iconos y botones de la interfaz.
- Interfaz basada en menús para realizar la composición.

Cabe mencionar que también existen otras ILNBDs que guían al usuario en la composición de consultas como Gingseng [Bernstein, 2005] y PANTO [Wang, 2007]; sin embargo, éstas son utilizadas para componer consultas en ontologías; por lo tanto, no se incluirán en este capítulo.

Para comprender cómo funcionan las herramientas ya mencionadas, en la siguiente Subsección se plantea el marco conceptual.

2.1 Marco conceptual

Este apartado contiene los principales conceptos que son involucrados en el desarrollo de las diversas interfaces que se presentan en la Subsección 2.2. De igual manera, dichos conceptos sirven para la interfaz desarrollada en este documento de tesis.

2.1.1 Bases de datos

Una base de datos (BDs) es un conjunto de datos relacionados entre sí. Por datos entendemos hechos conocidos que pueden registrarse y que tienen un significado implícito. Una base de datos tiene las siguientes propiedades implícitas [Elmasri, 1997]:

- Una base de datos representa algún aspecto del mundo real, en ocasiones llamado minimundo o universo de discurso. Las modificaciones del minimundo se reflejan en la base de datos.
- Una base de datos es un conjunto de datos lógicamente coherente, con cierto significado inherente. Una colección aleatoria de datos no puede considerarse propiamente una base de datos.
- Toda base de datos se diseña, construye y puebla con datos para un propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a dichos usuarios.

2.1.2 Lenguaje de consultas estructurado

SQL (Structured Query Language, por sus siglas en inglés) fue desarrollado por IBM, originalmente denominado SEQUEL, como parte del proyecto System R a principios de 1970. Hoy en día numerosos productos son compatibles con el lenguaje SQL, y se ha establecido como el lenguaje estándar para las bases de datos relacionales. La versión más reciente publicada por la ANSI (American National Standards Institute) es SQL:2008. SQL es una combinación de constructores del álgebra relacional y del cálculo relacional. Usando SQL es posible, además de definir la estructura de los datos, modificar los datos de la base de datos y especificar restricciones de seguridad [Silberschatz, 2006].

2.1.3 Lenguaje de definición de datos

Cuando se implementa una base de datos relacional, primero se debe definir la estructura de la misma con un sistema SMBD. Para tal efecto, los programadores usan un lenguaje de definición de datos (DDL, por sus siglas en inglés), cabe mencionar que para fines de este trabajo se hace uso del lenguaje de consulta SQL y su DDL correspondiente.

Haciendo uso del DDL se especifican los nombres de las tablas en la base de datos, se nombran y describen las columnas de esas tablas, definen los índices y se describen otras estructuras tales como restricciones y restricciones de seguridad [Kroenke, 2003].

En SQL, se hace uso de una colección de verbos imperativos cuyo fin es modificar el esquema de la BD añadiendo, cambiando, o borrando las tablas. Los estatutos del DDL pueden ser usados junto con otros estatutos en SQL.

Para ejemplificar el uso del DLL considérese el siguiente estatuto en SQL:

```
CREATE TABLE airport (
   airport_code        VARCHAR(50) PRIMARY KEY,
```

```
    airport_name      VARCHAR(50) NOT NULL,
    location          VARCHAR(50) NOT NULL,
    state_code        VARCHAR(50) NOT NULL,
    time_zone_code    VARCHAR(50) NOT NULL
);
```

El estatuto mencionado hace uso de la instrucción CREATE perteneciente al DDL. Dicha instrucción permite crear una tabla con cinco columnas dentro del esquema de la BD; asimismo, se puede hacer uso de las instrucciones Drop, Alter y Rename para destruir un objeto, modificarlo, o renombrarlo respectivamente.

2.1.4 Lenguaje de manipulación de datos

El lenguaje de manipulación de datos (DML, por sus siglas en inglés), permite la manipulación o procesamiento de los objetos definidos mediante el DDL [Date, 2001].

SQL permite hacer uso del DML para obtener y manipular información en una base de datos relacional. Las instrucciones en SQL que pueden ser usadas son las siguientes: Select, Insert into, Update y Delete from.

La instrucción usada principalmente en este proyecto de tesis es Select, la cual sirve para obtener un conjunto de resultados de una o más tablas. Considérese el siguiente ejemplo para ilustrar la sintaxis de dicha instrucción:

SELECT $T_i.C_i$ FROM T_i WHERE $T_i.C_j$ = <*valor*>

Donde $T_i.C_i$ es una columna de la base de datos, en cláusula From Ti es una tabla perteneciente a la base de datos, a partir de dicha tabla se obtendrá información. Por último, en la cláusula WHERE se especifican condiciones para restringir los resultados. Dicha condición se realiza asociando un valor a una columna de la BD.

2.1.3 ILNBDs

Una ILNBD es un sistema que permite al usuario acceder a la información almacenada en una base de datos formulando una solicitud en lenguaje natural [Androutsopoulos, 1995].

El resultado obtenido por una ILNBD puede ser presentado en dos formas; como una instrucción en SQL o como una respuesta en lenguaje natural. Este trabajo de investigación se centra solamente en la composición de consultas en SQL a semejanza de como se muestra en el flujo de una ILNBD (figura 2.1).

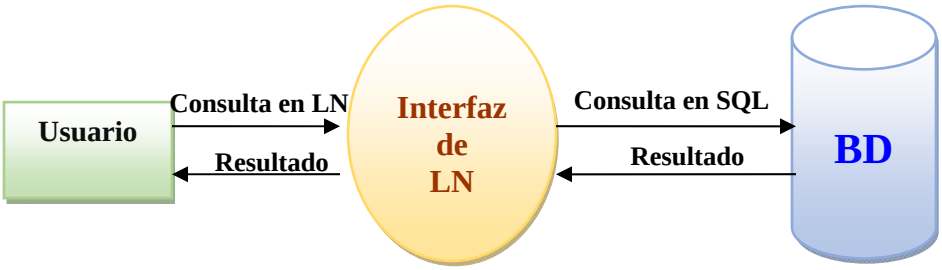


Figura 2.1. Flujo en una ILNBD

2.1.4 Composición y formulación de consultas

Composición

Según la Real Academia Española, componer se define como: *"formar de varias cosas una, juntándolas y colocándolas con cierto modo y orden"* [1]. La definición anterior se adopta en el presente trabajo para denotar la acción de construir gráficamente una consulta en SQL mediante un ciclo de preguntas del sistema y respuestas del usuario, permitiendo así que ésta se construya correctamente.

Formulación

De acuerdo a lo previsto por la Real Academia Española, formular se define como *"Reducir a términos claros y precisos un mandato, una proposición, una denuncia, etc."* [2]. El término formulación para fines de este trabajo es empleado para hacer referencia a la acción de escribir en LN una consulta a una ILNBD.

2.1.5 Diálogo

Un diálogo se puede describir como la interacción entre dos partes, en la cual la información es transferida entre las partes mediante un número de turnos secuenciales (un turno se refiere a la transferencia ininterrumpida de información de una parte a la otra) [Niesler, 2001].

Diálogo de aclaración

Un diálogo de aclaración es un ciclo interactivo entre un usuario y la ILNBD utilizado para especificar información faltante o ambigua en una consulta formulada por el usuario. La información faltante/ambigua es identificada por la ILNBD y posteriormente se ejecutan los ciclos de aclaración correspondientes.

Diálogo de composición

[1] http://lema.rae.es/drae/?val=componer
[2] http://lema.rae.es/drae/?val=formulaci%C3%B3n

Un diálogo de composición es la interacción que se efectúa entre un usuario y la ILNBD para guiarlo a través del proceso de composición, en el que se construye una consulta parte por parte hasta el punto en que ésta se encuentra lista para ser procesada por la ILNBD.

2.2 Trabajos relacionados

A continuación se presentan algunos trabajos relacionados con la composición de consultas sobre BDs relacionales.

2.2.1 CoBase - UCLA (1999)

En CoBase [Zhang, 1999], la formulación de consultas se basa en un grafo semántico (Figura 2.2), el cual proporciona un modelo semántico de la BD. El formulador de consultas requiere que el usuario especifique conceptos, atributos y valores relevantes para la consulta. Basado en la entrada de parámetros efectuada por el usuario, el sistema construye la consulta vía el grafo semántico. Cabe mencionar que el grafo semántico permite incorporar a éste relaciones definidas por el usuario además de las generadas por la interfaz.

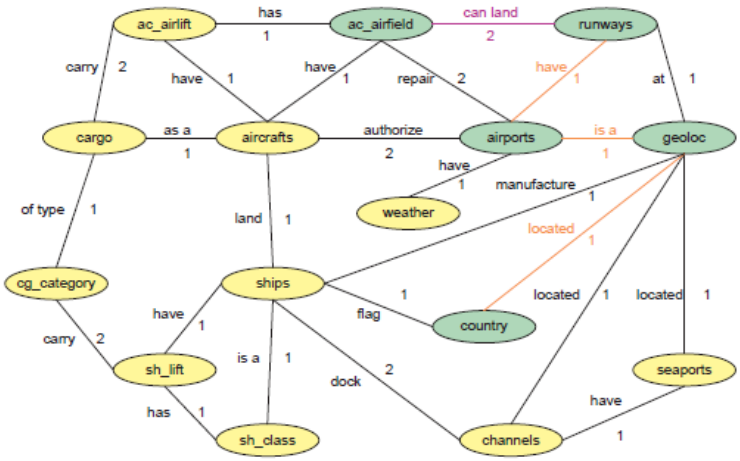


Figura 2.2. Grafo semántico de una BD de transporte

Los conceptos introducidos por el usuario se pueden dividir en tres partes: el aspecto de la consulta que corresponde a la cláusula Select de una consulta en SQL, las restricciones, y términos conceptuales que sirven como vínculo para los requerimientos de la consulta.

En la interfaz de especificación de conceptos y atributos, el usuario indica sobre qué tabla y columnas de la misma desea formular su consulta. Posteriormente el usuario debe especificar las restricciones de la consulta, en las que se debe indicar sobre qué atributos de qué tablas deben aplicarse las restricciones, también debe indicar el valor de la condición y elegir un operador para aplicar dicha condición. Una vez que el usuario introduce el aspecto de la consulta y sus restricciones, éste debe proceder a indicar una acción que asocie el aspecto de la consulta con sus restricciones.

Por último todos los datos introducidos por el usuario en los módulos anteriores son procesados por la ILNBD para generar un conjunto de consultas candidatas en SQL ordenadas de acuerdo a pruebas probabilísticas. Por lo anterior la interfaz solicita al usuario que elija una consulta de las obtenidas por la ILNBD, lo cual se realiza mediante un módulo de descripción de consultas formuladas presentado en la Figura 2.3, el cual genera una descripción de la consulta basada en los datos enlazados en el grafo semántico.

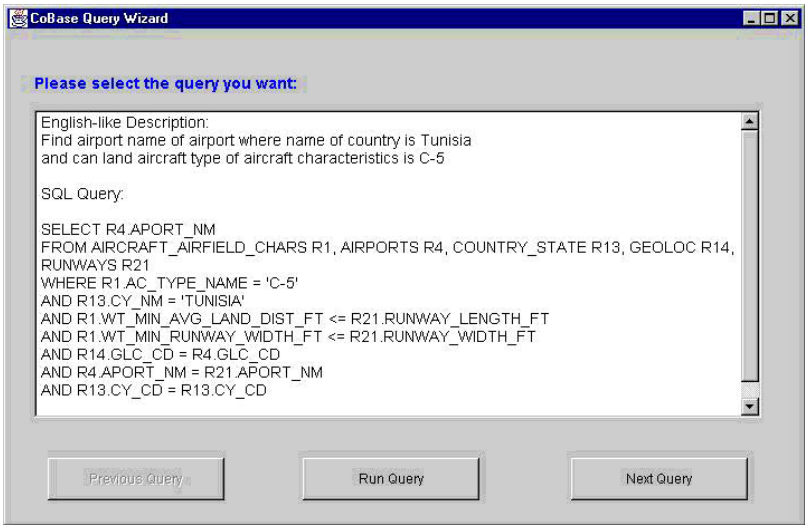


Figura 2.3. Consulta formulada en la interfaz de CoBase

2.2.2 Query Builder - University of Calgary (2004)

El trabajo descrito en [Little, 2004] presenta una interfaz de lenguaje natural dirigida a usuarios inexpertos. En esta interfaz el usuario formula su consulta, y la misma interfaz interactúa con el usuario para guiarlo sugiriéndole versiones modificadas de la consulta formulada. Esta interfaz puede conectarse con diferentes SMBDs como MySQL, Oracle, DB2 y Access. Esta interfaz ha sido probada con una BD de hockey.

La implementación de esta interfaz puede soportar preguntas basadas en números que involucren operadores de comparación. También soporta funciones de agregación. Sin embargo, una de las limitaciones de esta interfaz es el rango lingüístico sobre el que puede trabajar, ya que es solamente para el idioma inglés y además no soporta contracciones, como por ejemplo *don't* y *who've*.

Una de las características principales de esta interfaz es la retroalimentación que ofrece al usuario acerca de algún error que se detecte al momento de construir la consulta en SQL. Para estos casos, la interfaz es capaz de sugerir alternativas para guiar al usuario en la construcción de una consulta.

En la Figura 2.4 se puede observar la interfaz gráfica de Query Builder, la cual cuenta con controles para seleccionar diferentes bases de datos, una caja de texto para introducir una consulta en LN y una tabla para mostrar resultados.

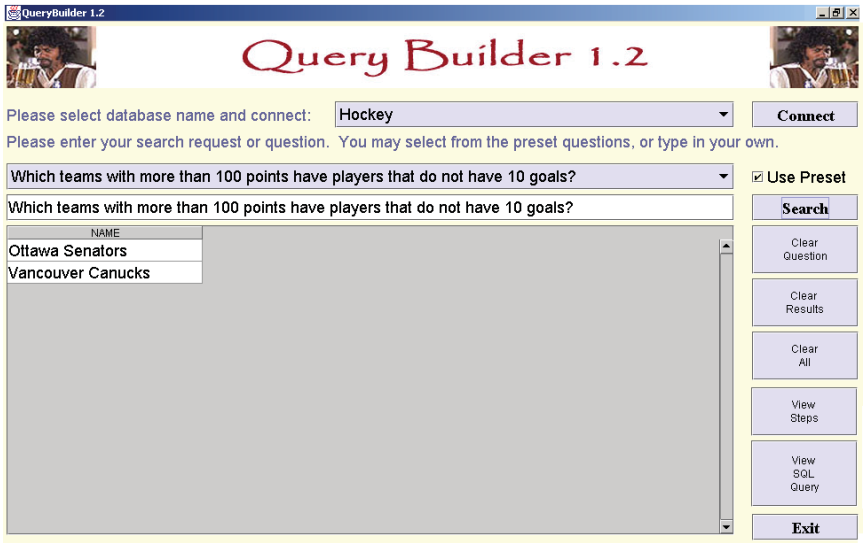


Figura 2.4. Interfaz principal de Query Builder

2.2.3 WYSIWYM - The Open University (2007)

WYSIWYM se basa en un método para inferir un conjunto de consultas posibles que se pueden aplicar a una base de datos, basándose en el grafo semántico descrito en [Zhang, 1999].

El sistema recibe como entrada un modelo semántico de la BD, que es generado manualmente por el usuario, y de manera automática genera las reglas, componentes y recursos que el sistema utiliza para traducir la consulta.

El sistema genera una plantilla de consulta por cada nodo conectado en el grafo semántico. Cada plantilla se construye emparejando el nodo actual con cada uno de los nodos a los que está vinculado, y en base a reglas gramaticales del idioma inglés se articula la plantilla. La interfaz presenta las plantillas de consulta en LN dejando entre corchetes la columna a la que posiblemente el usuario haga referencia. El usuario elige mediante un menú el tipo de pregunta que desea hacer y las columnas que desea referenciar en la consulta, como se muestra en la Figura 2.5.

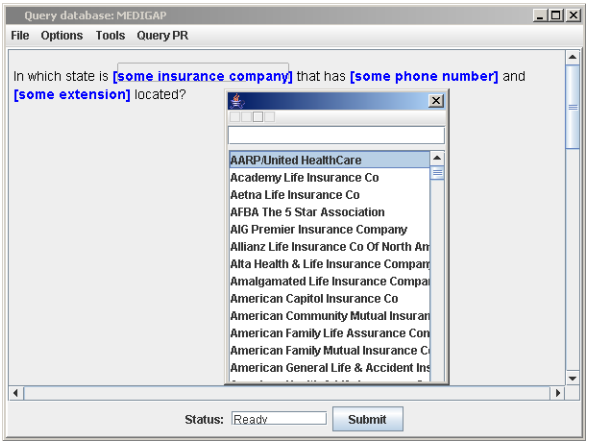


Figura 2.5. Editor de consultas de WYSIWYM

2.2.4 TAICHI – IBM T.J. Watson Research Center (2010)

En [Pan, 2010] se presenta una ILNBD que permite a los usuarios usar expresiones en lenguaje natural para componer consultas en un entorno visual, mejorando la usabilidad de una interfaz de consulta visual. Dicha herramienta se enfoca en ayudar a los usuarios inexpertos a obtener información de una BD rápidamente.

TAICHI usa un grafo de consulta para obtener la semántica de la consulta, el cual es mostrado en la interfaz de consulta visual; de esta forma, también define el alcance de las consultas que puede procesar. Dado que existe un mapeo directo desde los elementos del grafo de consulta hacia los elementos de la base de datos, es posible generar una consulta en SQL desde un grafo de consulta.

Esta herramienta cuenta con tres componentes: un intérprete de LN, un compositor visual (Figura 2.6) y un despachador. El intérprete traduce expresiones en LN a elementos del grafo de consulta. Dado un grafo de consulta, el compositor visual automáticamente genera una consulta visual o la actualiza.

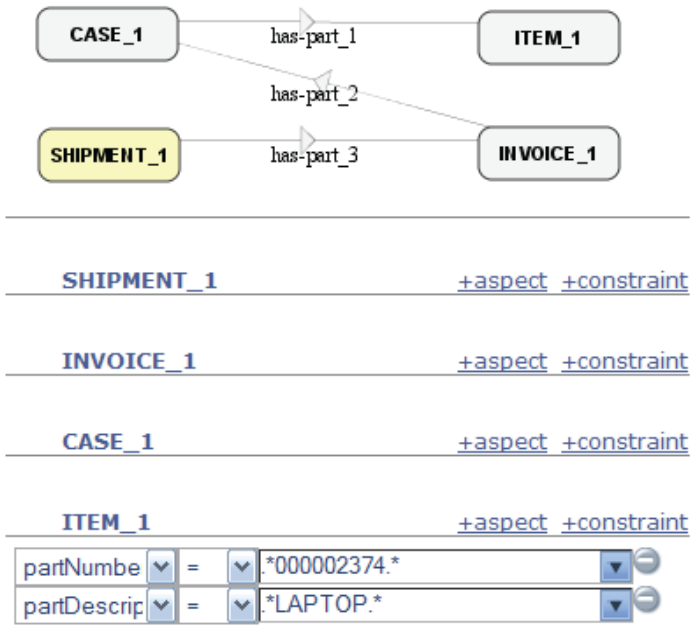


Figura 2.6. Ejemplo de una consulta visual en TAICHI

2.2.5 Microsoft Access QBE (2013)

Microsoft Access[3] es un motor de bases de datos relacionales de escritorio, el cual permite crear aplicaciones de BDs rápidamente mediante su interfaz gráfica. Los datos se almacenan automáticamente en una base de datos accesible mediante SQL. Al ser un producto para aplicaciones de escritorio, éste contiene herramientas diseñadas para obtener información de las BDs; es decir, posee herramientas que permiten al usuario componer consultas en SQL para obtener datos de una BD.

[3] http://products.office.com/en-us/access

Microsoft Access cuenta con una herramienta QBE que permite al usuario obtener información sobre las BDs, por medio de 3 pasos:
1.- Seleccionar las tablas involucradas en una consulta.
2.- Seleccionar las tablas y columnas involucradas en la cláusula Select.
3.- Seleccionar las tablas y columnas involucradas en la cláusula Where.

La interfaz gráfica de usuario de dicha herramienta se muestra en la Figura 2.7, donde se puede observar en la parte superior un segmento del esquema de BD y en la parte inferior, una cuadrícula que sirve para realizar la composición de una consulta.

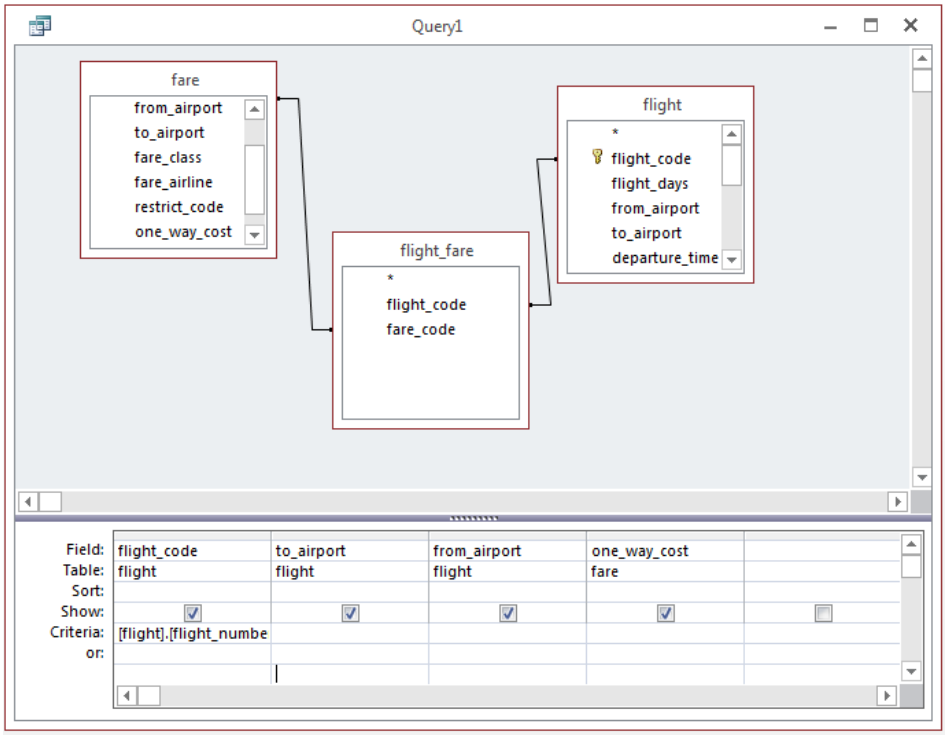


Figura 2.7. Interfaz QBE de MS Access

2.2.6 Conclusiones

Las interfaces presentadas en este capítulo muestran distintas técnicas de composición de consultas, las cuales permiten obtener información de una BD. Sin embargo, al utilizar una interfaz gráfica de usuario para componer consultas con un lenguaje de consulta formal, ésta puede limitar la cantidad de información que se puede llegar a obtener a partir de una BD.

Además, una interfaz de este tipo debe poder ser usada por usuarios inexpertos, es decir, usuarios no expertos en computación. Por lo tanto, es preferible evitar el uso de tecnicismos o términos difíciles de entender para los usuarios.

En la Tabla 2.1 se muestran las principales características que tienen las interfaces descritas previamente, así como la interfaz de composición objeto de esta tesis.

Tabla 2.1. Trabajos relacionados con la composición de consultas

Interfaz	Independencia de dominio	Métodos de composición	Explica el contenido de BD	Necesita conocimiento en SQL	Probado en BD complejas
CoBase	✓	Selección	×	✓	×
Query Builder	×	Menús	×	×	×
WYSIWYM	×	LN Plantillas	×	×	✓
TAICHI	✓	LN Drag & Drop	×	✓	×
MS Access	✓	Drag & Drop Exp. en SQL	×	×	✓
Interfaz propuesta	✓	Selección	✓	×	✓

Como se puede observar, las interfaces presentadas en la tabla anterior, no explican el contenido de la BD por lo que dificultan su uso por usuarios inexpertos; además, su independencia de dominio no ha sido comprobada.

Algunas interfaces no muestran el esquema de la BD debido a que su método de composición no lo requiere; sin embargo, al no mostrarlo el usuario no podría saber si la información requerida por su consulta se encuentra presente en la BD. Además, en la mayoría de las interfaces (por ejemplo la de la Figura 2.7), una base de datos como la de ATIS (Figura 2.8) sería extremadamente difícil de mostrar y de entender por usuarios inexpertos, en particular aquéllos que no tengan conocimiento previo de la BD.

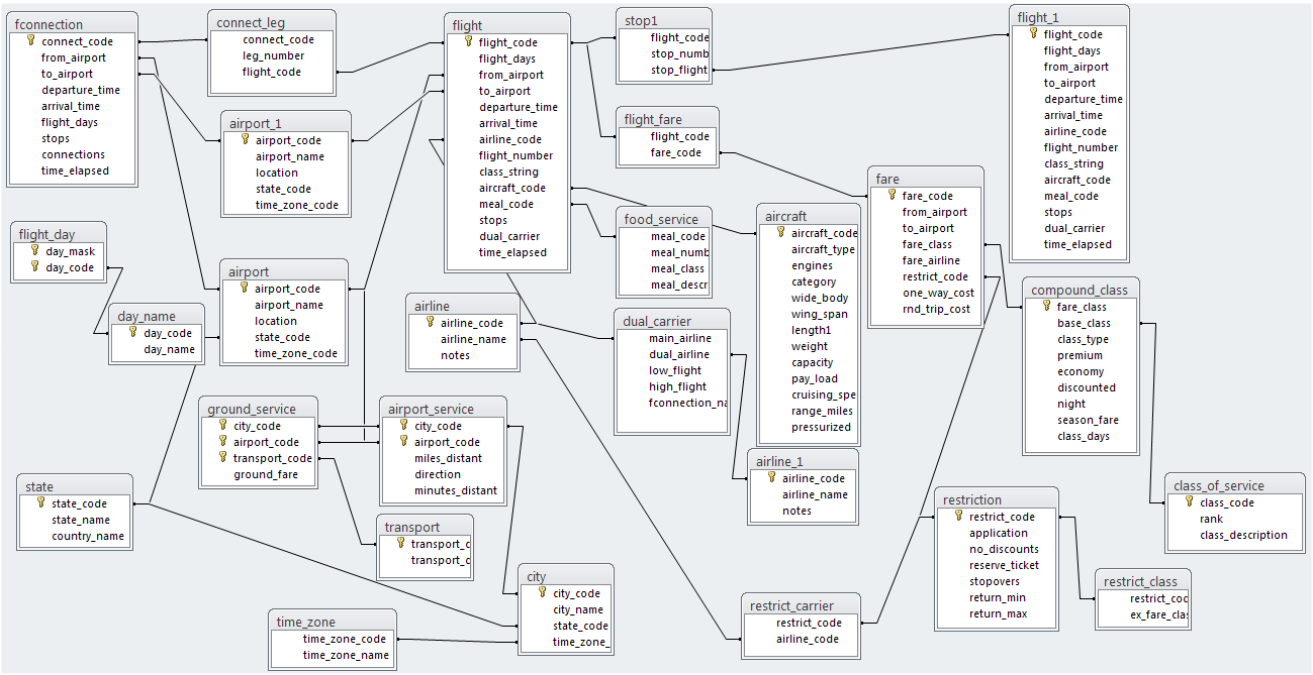


Figura 2.8. Esquema de la base de datos ATIS

Capítulo 3

Metodología de solución

La interfaz propuesta fue diseñada con el propósito de permitir a usuarios inexpertos componer consultas en SQL sobre bases de datos complejas. Un ejemplo de este tipo de bases de datos es ATIS (véase apéndice A), la cual cuenta con un total de 27 tablas y 123 columnas.

Para lograr tal fin, la interfaz de composición de consultas obtiene información sobre el esquema de BD a partir del diccionario de información semántica (DIS) [Aguirre, 2014]. Posteriormente, la interfaz muestra la información obtenida en un árbol de composición para que el usuario pueda hacer uso de dicha información. Asimismo, la interfaz permite al usuario componer su consulta de manera intuitiva realizando la selección de los datos que desee conocer sin necesidad de buscarlos directamente en el esquema de la BD.

Sin embargo, para que la interfaz contenga la información suficiente sobre el esquema de la BD, la interfaz debe ser configurada previamente por el administrador de BD. El administrador proveerá a la interfaz información relacionada con los datos contenidos dentro de la BD (descripciones para las tablas y columnas, clasificaciones de tablas, tipos de datos para las columnas, etc.), y de esta forma dichos datos se presentarán de manera más clara al usuario.

3.1 Arquitectura propuesta

En la Figura 3.1 se puede apreciar la arquitectura propuesta, la cual se basa en cuatro etapas involucradas en el procesamiento de una consulta: selección del tema de interés, selección de elementos de interés, especificación de condiciones de búsqueda, y por último, vista previa y resultados. Las primeras tres etapas forman parte del proceso de composición de la consulta, mientras que la última muestra el resultado obtenido de dicho proceso.

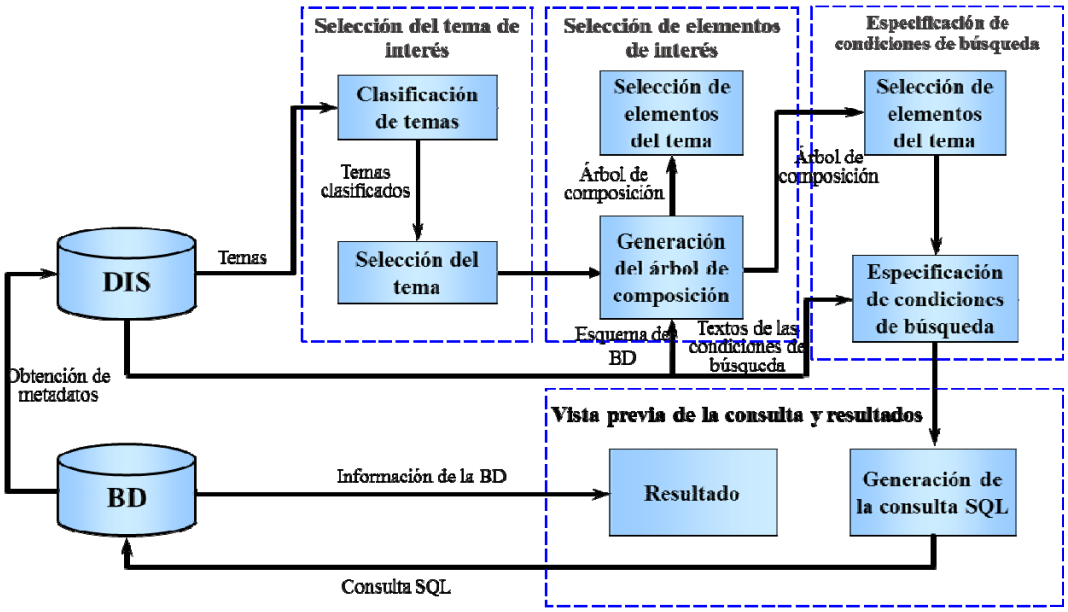


Figura 3.1. Arquitectura propuesta de la interfaz de composición

La arquitectura propuesta guía al usuario por el proceso de composición, en donde la primera etapa es la selección del tema de interés. En esta etapa la interfaz toma información sobre las tablas existentes en la BD, las clasifica de acuerdo a su tipo y las muestra al usuario como una lista de temas clasificados, en donde el usuario deberá seleccionar un tema para basar su consulta sobre éste.

En la segunda etapa, la interfaz genera un árbol de composición haciendo uso del DIS, a partir del cual el usuario selecciona uno o varios elementos de éste. Dichos elementos serán usados para construir la cláusula Select de la consulta en SQL.

En la tercera etapa, se emplea el mismo árbol de composición para especificar las condiciones de búsqueda por las cuales la interfaz restringirá los resultados. Estas condiciones de búsqueda son empleadas para construir la cláusula Where de la consulta en SQL.

Posteriormente, la interfaz genera una consulta en SQL a partir de los elementos seleccionados por el usuario y las condiciones de búsqueda especificadas. Finalmente, la interfaz envía dicha consulta en SQL al SMBD para obtener los resultados correspondientes y muestra el resultado al usuario.

3.2 Clasificación de las tablas de una base de datos

La clasificación de tablas es usada por la interfaz para agrupar las tablas de la BD de acuerdo a su relevancia para la composición de consultas. Además, la clasificación de tablas permite obtener información sobre el esquema de BD para construir el árbol de composición, el cual es el mecanismo más importante de la interfaz. La Tabla 3.1 muestra en orden de relevancia los diferentes tipos de tablas que son usadas para la composición de consultas.

Tabla 3.1. Clasificación de tablas

Tipo de tabla	Descripción
1. Tablas base	Tablas principales que almacenan información que se usa frecuentemente para realizar consultas sobre la BD.
2. Vistas	Tablas virtuales que son obtenidas a través de una consulta que involucra tablas base y son usadas para obtener información que no puede ser obtenida directamente de las tablas base.
3. Catálogos	Tablas que son usadas principalmente para obtener una descripción en LN de una columna a partir de una clave o código.
4. Relaciones M a N	Tablas que contienen llaves foráneas que pertenecen a otras tablas (T_j y T_j) y se usan para implementar relaciones M a N entre $T_{i\,y}\,T_j$.
5. Tablas satélite	Tablas que se encuentran desconectadas del resto de las tablas. Estas tablas son usadas por los procesos internos de las aplicaciones que usan la BD.

De la tabla anterior, los tipos de tabla con mayor relevancia para la composición de consultas son las tablas base, vistas y catálogos; sin embargo, en este trabajo no se implementa la funcionalidad para trabajar con Vistas.

3.3 Tipos de consultas a tratar

En base a la clasificación de tablas propuesta, se han considerado nueve tipos de consultas que involucran diferentes tipos de tablas. Como se puede observar en la Tabla 3.2, las consultas que se abordan en este trabajo son del tipo 1 al 5; es decir, consultas que involucran desde una tabla base, hasta consultas que involucran tres tablas base conectadas por medio de dos tablas que implementan una relación *muchos* a *muchos* y que no impliquen la ejecución de subconsultas.

Tabla 3.2. Tipos de consultas

Tipo de consulta	No. de tablas base	No. de tablas M a N	Subconsultas
Tipo 1	1	0	No
Tipo 2	2	0	No
Tipo 3	3	0	No
Tipo 4	2	1	No
Tipo 5	3	2	No
Tipo 6	2	0	Si
Tipo 7	3	0	Si
Tipo 8	2	1	Si
Tipo 9	3	2	Si

3.4 Árbol de composición

El árbol de composición es una estructura gráfica que se asemeja a un explorador de archivos. Esta estructura se usa para enfocar la atención del usuario sobre un fragmento de la BD. Dicha estructura permite visualizar el esquema de BD como un árbol que contiene los elementos del tema principal (columnas de la tabla raíz) y temas relacionados (tablas relacionadas a la tabla raíz). Como cualquier árbol, éste debe ser generado a partir de una raíz, en este caso, una tabla de la BD.

3.4.1 Construcción

La información utilizada por la interfaz para construir el árbol de composición se obtiene del DIS (véase Figura 4.1). Dicha información se menciona a continuación:

- Por cada tabla de la BD:
  - Tipo de tabla.
  - Descripción de la tabla en LN.
  - Descripciones de las columnas en LN.
  - Relaciones entre la tabla y otras tablas.
- Por cada relación entre tablas:
  - Tipo de relación.

A partir de esta información, la interfaz puede construir el árbol de composición, el cual tiene como raíz una tabla de cualquier tipo a excepción de las tablas satélite, las cuales la mayoría de las veces no tienen conexión con otras tablas. El árbol de composición mostrará solamente descripciones en LN de las tablas y columnas de la BD, manteniendo ocultos los nombres de las tablas y columnas.

La construcción del árbol de composición es efectuada por la interfaz en la segunda fase del proceso de composición (selección de los elementos de interés), a partir de una tabla seleccionada por el usuario en la primera fase.

En el Algoritmo 5.1 se presenta el pseudocódigo para la construcción del árbol de composición, donde *CT* es el árbol de composición, *n* es el nodo de relación que hace referencia a una tabla, *R* es un conjunto de tablas que se encuentran relacionadas con la tabla *t*. La construcción consiste en inicializar *CT* insertando la tabla raíz T_r y aplicar la función recursiva insertarRelaciones (mostrada en la línea 1), la cual requiere una tabla como entrada. Posteriormente la interfaz obtiene las tablas *R*, las cuales se encuentran relacionadas con *t* y por cada tabla *r*, evalúa su tipo procediendo de la siguiente manera:

- Si la tabla relacionada es una tabla base, aplica la función recursiva para insertar las columnas y relaciones correspondientes en el árbol de composición *CT* (línea 8).
- Si la tabla relacionada es una tabla catálogo, se insertan los nodos correspondientes a las columnas de ésta en el nodo padre de la tabla (línea 11).
- Si la tabla relacionada es una tabla de relación M a N, se obtiene la tabla relacionada a ésta (línea 14) y se aplica la función para insertar las tablas relacionadas (línea 15), ocultando la tabla de relación *M* a *N*.

Algoritmo 5.1. Pseudocódigo para construir un árbol de composición

```
1:  insertarRelaciones(t)
2:      p  //Nodo padre de t
3:      n ← insertarNodoRelacion(CT_p, t)
4:      insertarNodoColumna(CT_n, t)
5:      R ← obtenerTablasRelacionadas(t)
6:      for each r from R do
7:          if esTablaBase (r)
8:              insertarRelaciones(r)
9:          Endif
```

```
10:          if esCatalogo(r)
11:              insertarNodosColumna(CT_n, r)
12:          Endif
13:          if esMaN (r)
14:              r' ← obtenerTabla(r)
15:              insertarRelaciones(r')
16:          Endif
17:      Endfor
18:  End
```

En la Figura 3.2 se muestra un fragmento del árbol de composición construido para la base de datos ATIS, cuya raíz es la tabla *flight*, y cuya descripción es *vuelo de un aeropuerto a otro*. Nótese que el primer nodo del árbol indica la raíz o tema principal, mientras que los nodos hijos de éste representan elementos de éste (en color verde), o temas relacionados (en color azul).

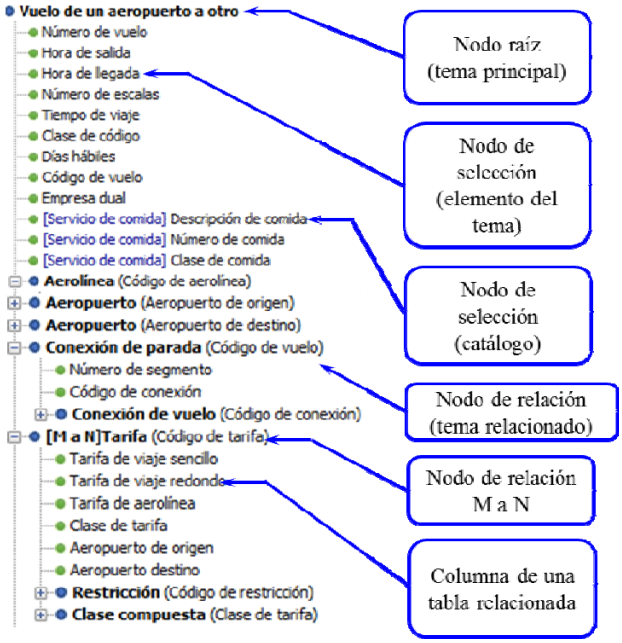


Figura 3.2. Ejemplo de un árbol de composición

3.4.2 Características

El árbol de composición se implementa en la interfaz de composición con el propósito de permitir al usuario realizar selecciones sobre el mismo en dos fases diferentes del proceso de composición: las fases de selección de elementos de interés y definición de condiciones de búsqueda.

En la etapa de selección de elementos de interés, el árbol de consulta se emplea para que el usuario seleccione los nodos referentes a los elementos que le interesa conocer de la BD, y así pueda añadirlos a una lista donde quedarán almacenadas dichas selecciones.

En la etapa de definición de condiciones de búsqueda, el árbol de consulta es usado por el usuario para elegir sobre qué elemento desea especificar una condición para restringir los resultados de la consulta.

Además, el árbol de composición cuenta con las siguientes características que le permiten al usuario realizar una selección fácil e intuitiva de los elementos:

- Impide que el usuario seleccione nodos de relación, nodos de relación M a N y el nodo raíz, debido a que estos nodos hacen referencia a tablas y no a columnas.
- Muestra las columnas de las tablas catálogo como si fueran columnas pertenecientes a la tabla a la que están relacionadas las tablas catálogo.
- En la fase de Selección de elementos de interés, impide que el usuario seleccione y añada a la lista dos veces el mismo elemento.
- Cuando una tabla base (*T*) contiene columnas que son llaves foráneas (*Fk*), éstas se ocultan. Las columnas que son llaves primarias y realizan una conexión con *T* (mediante *Fk*) son mostradas en el nodo correspondiente. Esto se realiza con el fin de que el usuario tenga una mejor visualización de las columnas de la BD. Además, evita que se confunda con el despliegue de dos columnas con la misma descripción en dos tablas diferentes.
- El número máximo de niveles del árbol es de 4 para evitar ciclos (que se vuelva a mostrar un fragmento del esquema de BD). Esto además, limita la profundidad del árbol facilitando así la navegación en el contenido del mismo.

Capítulo 4

Interfaz de composición

La interfaz de composición es una interfaz gráfica diseñada con el fin de ayudar a los usuarios a componer consultas en SQL sobre una BD sin necesidad de conocer el esquema de la misma. Para ello, la interfaz cuenta con cuatro ventanas que guían al usuario a través del proceso de composición de consulta.

Sin embargo, aunque se supone que el usuario no conoce el esquema de BD y tampoco tiene conocimientos sobre SQL, es importante que tenga una idea clara de los datos que desea conocer de la BD, ya que las consultas compuestas por la interfaz dependen mucho del conocimiento del usuario sobre lo que desea conocer.

4.1 Conceptos para la composición de consultas

El proceso de composición de consultas es llevado a cabo por el usuario por medio de la interfaz de composición. Antes de comenzar a componer una consulta, el usuario deberá tener en cuenta los conceptos presentados en la Tabla 4.1.

Tabla 4.1. Conceptos involucrados en la composición de consultas

Concepto	Descripción	Relación con la BD
Tema de interés	Tema sobre el cual el usuario desea conocer. Nota: la consulta gira en torno a este tema.	Tabla que será la raíz del árbol de composición.
Elementos específicos del tema de interés	Información específica que desea obtener el usuario acerca del tema de interés y sus temas relacionados.	Columnas de la tabla raíz y tablas relacionadas, las cuales serán usadas para generar la cláusula Select.
Condiciones de búsqueda	Elementos y valores pertenecientes al tema de interés que sirven para acotar la información que se requiere obtener.	Columnas de la tabla raíz y tablas relacionadas con operadores de comparación y valores, los cuales serán usados para generar la cláusula Where.

Una vez que los conceptos mencionados en la Tabla 4.1 hayan sido identificados por el usuario en la consulta que desea componer, podrá hacer uso de la interfaz de composición para buscar dichos elementos en el árbol de composición.

4.2 Selección del tema de interés

La primera ventana que muestra la interfaz de composición se usa para especificar el tema de interés de la consulta (tabla raíz del árbol de composición), la cual servirá para construir el árbol de composición y componer la consulta basada en el tema elegido. Los componentes de la ventana con sus respectivas descripciones se listan en la Tabla 4.2.

Tabla 4.2. Componentes de la ventana Selección del tema de interés

Componente	Descripción
1. Barra de menú	Menú que contiene las opciones Archivo, Herramientas y Ayuda. La primera puede ser usada cancelar el proceso de composición de consulta, mientras que la segunda es usada por el administrador de BD para configurar la interfaz y la tercera opción sirve para mostrar el manual de usuario.
2. Lista de temas	Muestra los temas (tablas) que contiene la BD agrupados por tipo; inicialmente muestra las tablas base; además, no permite seleccionar más de una tabla.
3. Botones de navegación de temas	Son usados para cambiar el contenido de la lista de temas y navegar entre los distintos tipos de tablas.
4. Botón Continuar	Se usa para continuar con el proceso de composición. El botón se encuentra deshabilitado hasta que se realiza una selección válida sobre la lista de temas.
5. Botón Cancelar	Cancela el proceso de composición cerrando la interfaz de composición.

Los componentes mencionados en la Tabla 4.2 pueden ser identificados en la Figura 4.1, y se relacionan mediante el número de componente.

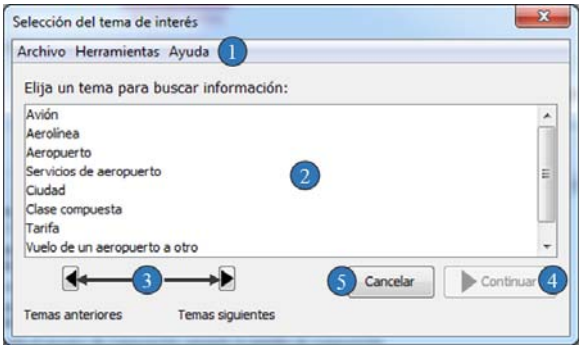


Figura 4.1. Componentes de la ventana Selección del tema de interés

El componente más importante de la ventana mostrada en la Figura 4.1 es la *Lista de temas*, la cual presenta una lista con las descripciones de las tablas contenidas en la BD. Dicha lista muestra un tipo de tabla a la vez, comenzando por las tablas de mayor relevancia para la composición de consultas. Por ejemplo, la clasificación de tablas para la base de datos ATIS es presentada en la Tabla 4.3, en la cual se pueden observar cuatro tipos diferentes de tablas.

Tabla 4.3. Clasificación de tablas para la base de datos ATIS

Tablas base	Catálogos	Relaciones M a N	Tablas satélite
Avión	Clase de servicio	Vuelo – tarifa	Descripción de código
Aerolínea	Servicio de comida	Restricción de empresa	Día
Aeropuerto	Clases de restricción	Conexión de parada	Días de vuelo
Servicios de aeropuerto	Estado	Empresa doble	Nombres de mes
Ciudad	Zona horaria	Servicio terrestre	Intervalo de tiempo
Clase compuesta	Servicio de transporte del aeropuerto	Escala de vuelo	
Tarifa			
Conexión de vuelo			
Vuelo de un aeropuerto a otro			

Las tablas clasificadas podrán ser mostradas en la lista haciendo uso de los botones de navegación de temas ubicados en la parte inferior de la ventana. La lista comenzará mostrando las tablas base, posteriormente si el usuario utiliza los botones de navegación, la interfaz mostrará las tablas catálogo, y así sucesivamente de acuerdo al orden como se muestra en la Tabla 4.3.

4.3 Selección de elementos de interés

Esta ventana es usada por el usuario para elegir los elementos del tema de interés de los cuales desea obtener información. Los componentes que integran esta ventana se muestran en la Tabla 4.4.

Tabla 4.4. Componentes de la ventana Selección de elementos de interés

Componente	Descripción
1. Árbol de composición	Se usa para representar el esquema de BD a partir de una tabla. Pueden realizarse una o más selecciones sobre el mismo, pero sólo puede seleccionarse un elemento del árbol a la vez.
2. Seguimiento de consulta	Sección de la ventana que se usa para mostrar las selecciones que realiza el usuario en el proceso de composición de consultas.
3. Lista de elementos de interés	Lista usada para contener los elementos seleccionados en el árbol de consulta. Dichos elementos indican los datos que el usuario desea conocer de la BD. No permite la existencia de elementos duplicados.
4. Añadir elementos	Botón usado para añadir un elemento del árbol de composición a la *Lista de elementos de interés*.
5. Eliminar elementos	Botón usado para eliminar elementos de la *Lista de elementos de interés*.
6. Botón Continuar	Botón usado para continuar con el proceso de composición. Se encuentra deshabilitado hasta añadir un elemento a *la Lista de elementos de interés.*
7. Botón Cancelar	Cancela el proceso de composición cerrando la interfaz de composición.

Los componentes mencionados en la tabla anterior pueden ser identificados en la Figura 4.2, donde el usuario debe seleccionar uno a uno los elementos que desee conocer. Posteriormente se añaden dichos elementos a la *Lista de elementos de interés* mediante el botón *Añadir elemento.*

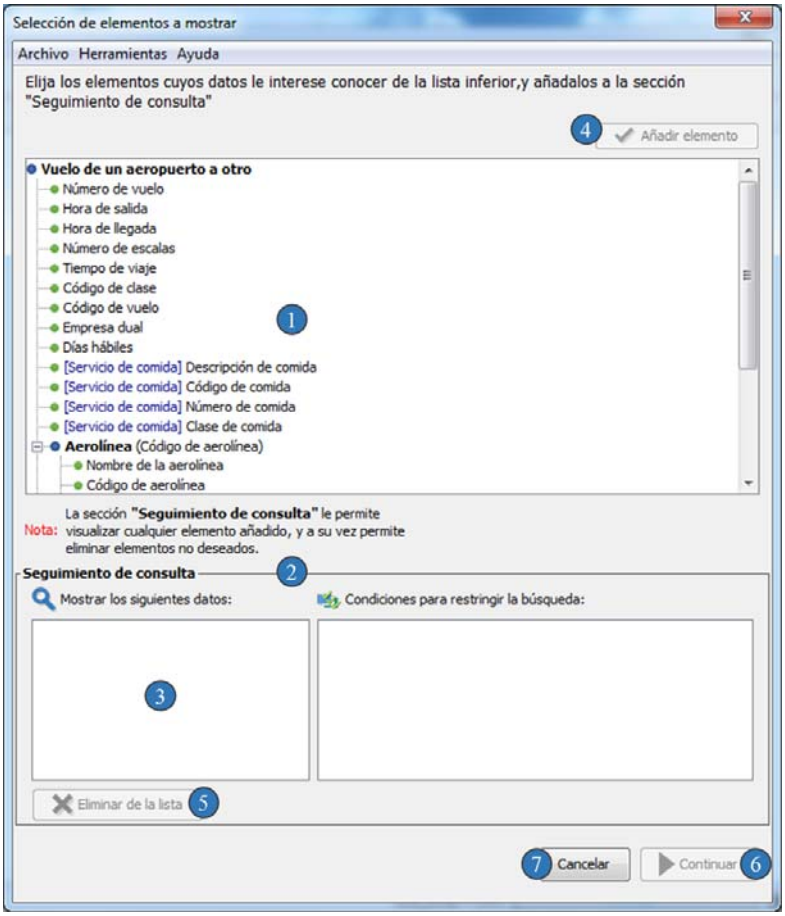


Figura 4.2. Componentes de la ventana Selección de elementos de interés

Por cada elemento añadido a la lista de elementos de interés, la interfaz guarda un vector de nodos que representa la ruta seguida desde la raíz del árbol hacia el elemento seleccionado. Cada nodo es representado como un vector de cuatro posiciones, en donde la primer posición almacena la descripción de la tabla a la que pertenece el nodo, la segunda posición almacena la descripción de la columna que corresponde al nodo, la tercera posición almacena la relación existente entre el nodo anterior y éste, la cuarta posición representa el tipo de tabla a la que pertenece el nodo. La Figura 4.3 muestra cómo se almacena la información concerniente a la generación de una ruta.

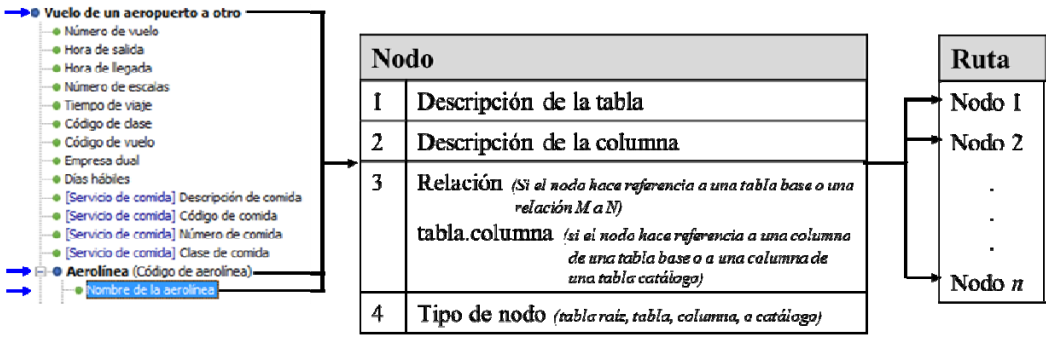
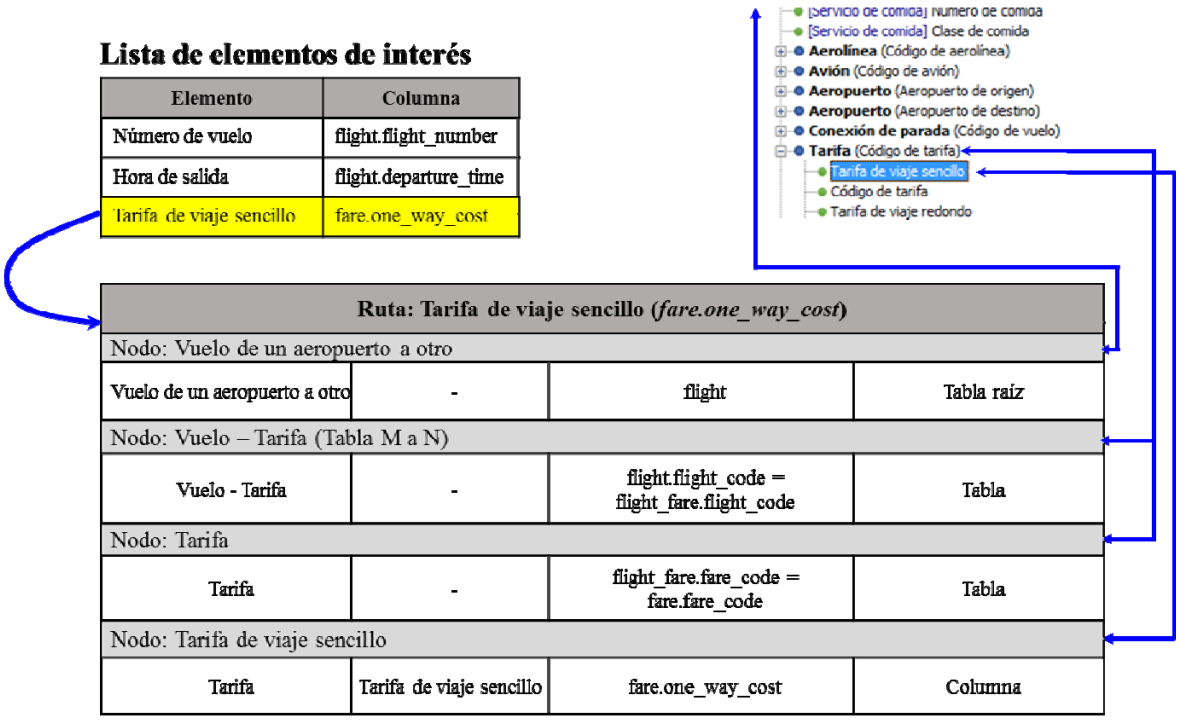


Figura 4.3. Construcción de la ruta de un nodo seleccionado

En la figura 4.4 se muestra un ejemplo de cómo se realiza la construcción de una ruta para un elemento de interés. En dicho ejemplo se observa un conjunto de tres elementos de interés, los cuales

fueron seleccionados por el usuario. La ruta del elemento *Tarifa de viaje sencillo* consta de cuatro nodos: *Vuelo de un aeropuerto a otro* (Tabla raíz), *Vuelo – Tarifa* (Tabla M a N), *Tarifa* (Tabla), *Tarifa de viaje sencillo* (Nodo columna).



Lista de elementos de interés

Elemento	Columna
Número de vuelo	flight.flight_number
Hora de salida	flight.departure_time
Tarifa de viaje sencillo	fare.one_way_cost

Ruta: Tarifa de viaje sencillo (*fare.one_way_cost*)			
Nodo: Vuelo de un aeropuerto a otro			
Vuelo de un aeropuerto a otro	-	flight	Tabla raíz
Nodo: Vuelo – Tarifa (Tabla M a N)			
Vuelo - Tarifa	-	flight.flight_code = flight_fare.flight_code	Tabla
Nodo: Tarifa			
Tarifa	-	flight_fare.fare_code = fare.fare_code	Tabla
Nodo: Tarifa de viaje sencillo			
Tarifa	Tarifa de viaje sencillo	fare.one_way_cost	Columna

Figura 4.4. Ejemplo de la construcción de la ruta de un elemento de interés

Una vez que el usuario haya terminado de añadir los elementos a la lista de elementos de interés, la interfaz tendrá un conjunto de vectores que representan las rutas de cada elemento de interés, éstas se almacenan en un vector de *m* posiciones, donde *m* es el número total de elementos añadidos por el usuario, tal como se muestra en la Figura 4.5.

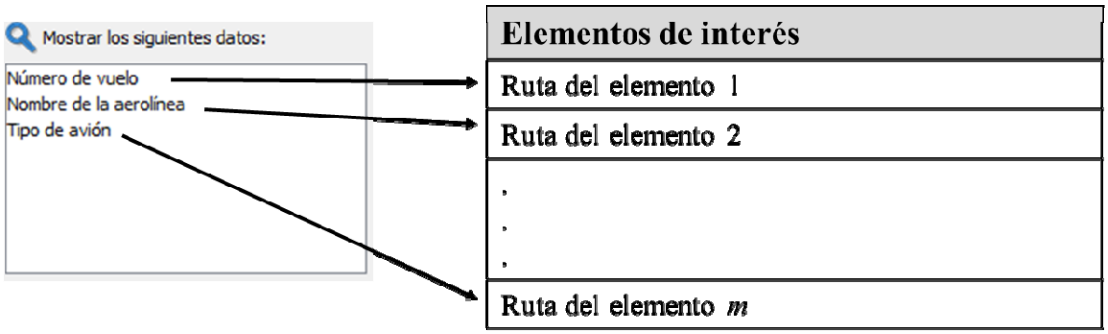


Figura 4.5. Vector de elementos de interés

Por último, la interfaz preguntará al usuario si desea especificar condiciones de búsqueda para discriminar datos. En caso de que el usuario conteste afirmativamente, se mostrará la ventana *Definición de condiciones de búsqueda*; de otro modo se mostrará la ventana *Vista previa y resultados* de la consulta y no se especificarán condiciones de búsqueda para la consulta.

4.4 Definición de condiciones de búsqueda

Esta ventana es la última del proceso de composición, en donde el usuario define las condiciones de búsqueda que permiten discriminar datos que no son de interés para el usuario.

En la Tabla 4.5 se describe el uso de cada uno de los componentes que conforman esta ventana, y en la Figura 4.5 se pueden identificar dichos componentes por medio de sus respectivos números.

Tabla 4.5. Componentes de la ventana Definición de condiciones de búsqueda

Componente	Descripción
1. Árbol de composición	Se usa para representar el esquema de BD a partir de una tabla. Pueden realizarse una o más selecciones sobre el mismo, pero sólo puede seleccionarse un elemento del árbol a la vez.
2. Selección del árbol de composición	Muestra el elemento que se ha seleccionado en el árbol de composición, y sirve para visualizar sobre cuál elemento se especificará la condición de búsqueda.
3. Lista de operadores de comparación	Lista usada para relacionar el elemento seleccionado en el árbol de composición con el valor de la condición de búsqueda.
4. Valor de la condición	Caja de texto usada para escribir el valor de la condición de búsqueda.
5. Botón de ayuda	Se usa para mostrar una ventana emergente que contiene información sobre el elemento seleccionado en el árbol de composición.
6. Lista de condiciones de búsqueda	Lista que almacena las condiciones de búsqueda definidas por el usuario.
7. Añadir condición de búsqueda	Una vez que una condición de búsqueda contiene un operador de comparación y un valor, ésta puede ser añadida a la lista de condiciones de búsqueda mediante este botón.
8. Eliminar condición de búsqueda	Elimina la condición de búsqueda seleccionada en la *Lista de condiciones de búsqueda*.
9. Botón Continuar	Botón usado para continuar con el proceso de composición; se encuentra deshabilitado hasta añadir un elemento a la *Lista de condiciones de búsqueda*.
10. Botón Cancelar	Cancela el proceso de composición cerrando la interfaz de composición.

Esta ventana utiliza el mismo árbol de composición que la ventana *Selección de elementos de interés*, con la diferencia de que esta ventana se usa para adjuntar una condición a un elemento del árbol.

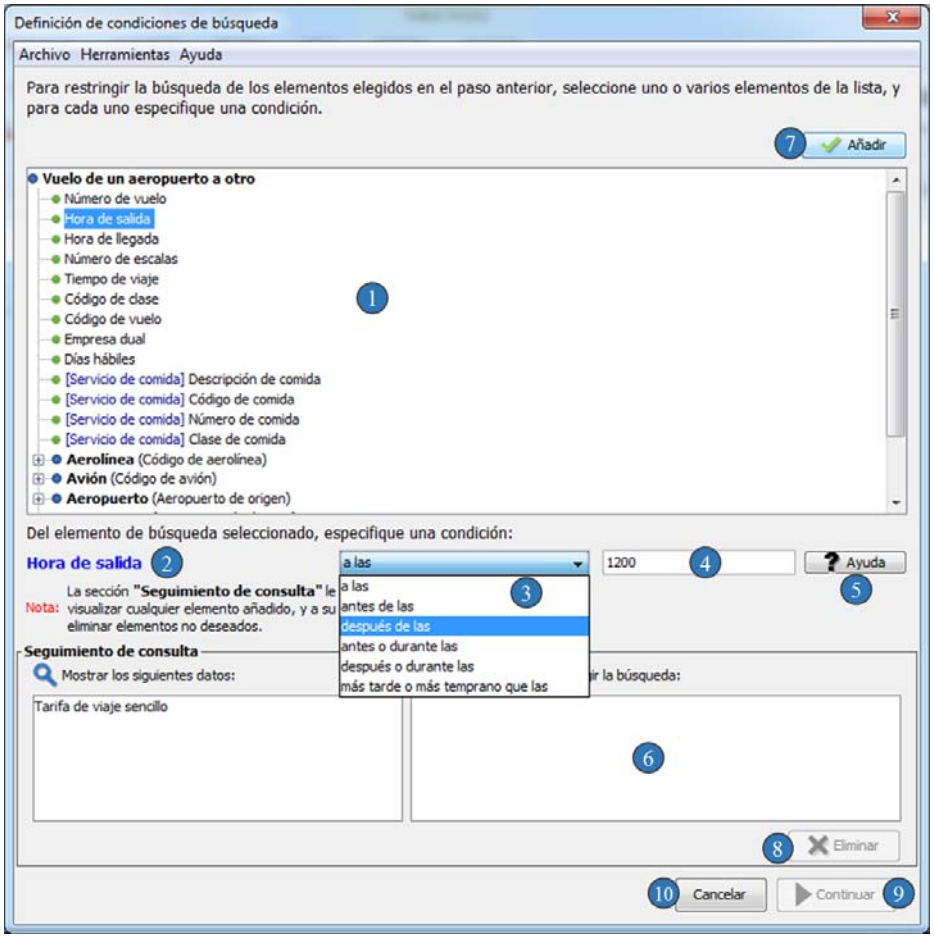


Figura 4.6. Componentes de la ventana Definición de condiciones de búsqueda

Dado un nodo seleccionado en el árbol de composición, la interfaz obtiene del DIS un conjunto de descripciones referentes a los operadores de comparación que pueden ser usados para el nodo, tomando en cuenta el tipo de dato al cual pertenece, y llena la lista de operadores de comparación con dichas descripciones (Figura 4.7).

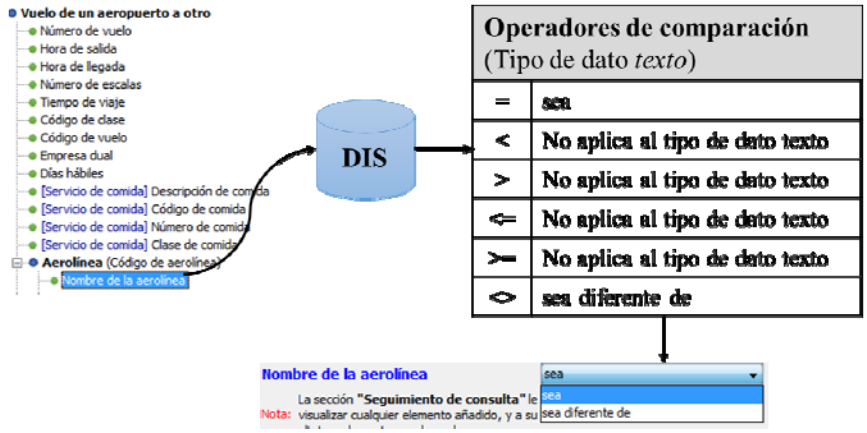


Figura 4.7. Llenado de la lista de operadores de comparación

Ya seleccionado uno de los operadores de comparación de la lista de operadores, el usuario escribe el valor de la condición de búsqueda. En caso de no saber qué formato debe tener el valor para un elemento, puede hacer clic en el botón de ayuda, el cual mostrará la información relevante para dicho elemento.

Cuando se haya seleccionado un elemento del árbol de composición, escrito un valor para este elemento y seleccionado un operador de comparación de la lista de operadores, se habrá definido una condición de búsqueda y podrá ser añadida a la lista de condiciones de búsqueda mediante el botón *Añadir* que se encuentra en la parte superior derecha de la ventana.

Siempre que un usuario añada una condición de búsqueda a la lista, la interfaz almacena la condición de búsqueda como un vector de cinco posiciones, las cuales se detallan a continuación:

- En la primera posición, se almacena un vector que representa el nodo seleccionado, tal y como se explicó en la Subsección 4.3.
- En la segunda posición, se almacena el operador de comparación correspondiente a la descripción seleccionada en la lista de operadores.
- En la tercera posición, se almacena la descripción del operador.
- En la cuarta posición, se almacena el valor de la condición tomando dicho valor de la caja de texto indicada con el número 4 en la Figura 4.6.
- En la quinta posición, se almacena la ruta que existe entre el nodo raíz y el nodo seleccionado, tal y como se explicó en la Subsección 4.3.

En la Figura 4.8 se ilustra la forma en la que se conforma una condición de búsqueda definida mediante la interfaz.

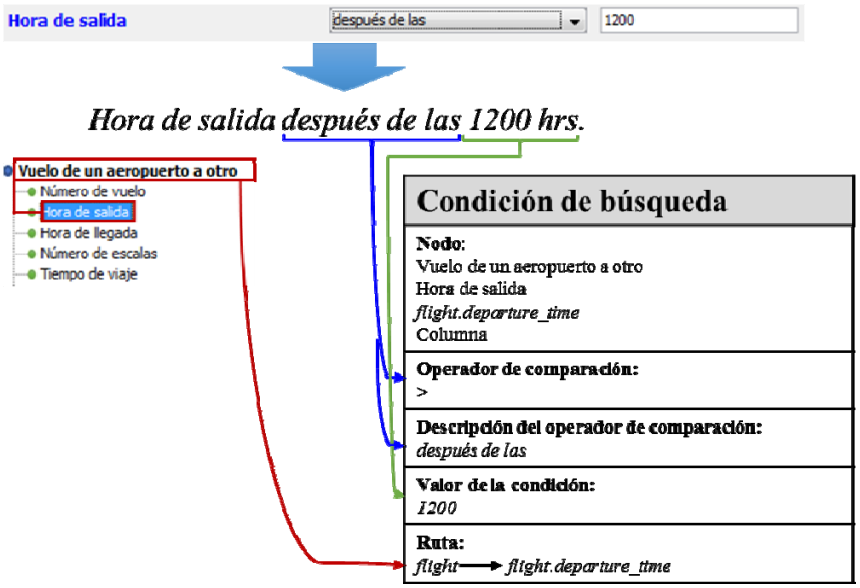


Figura 4.8. Definición de una condición de búsqueda

Una vez que el usuario termine de definir las condiciones de búsqueda, la interfaz tendrá una lista de condiciones (Figura 4.9), las cuales serán utilizadas por la interfaz para generar la cláusula Where de la consulta.

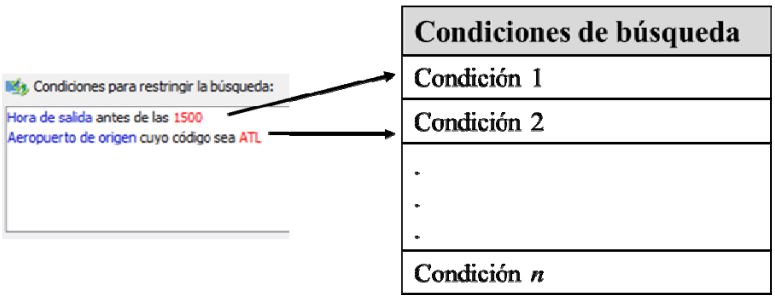


Figura 4.9. Lista de condiciones de búsqueda

Ya terminados los tres pasos del proceso de composición, la interfaz podrá generar la consulta en SQL a partir de los elementos de interés y las condiciones de búsqueda ya definidos. La generación de la consulta en SQL se describe en la siguiente Subsección.

4.5 Generación de la consulta en SQL

El objetivo principal de la interfaz propuesta en este proyecto de tesis consiste en la generación de una consulta en SQL. Dicha consulta se constituye de tres partes: una cláusula *Select*, una cláusula *From* y una cláusula *Where*.

Para la cláusula *Select*, a partir de la lista de elementos de interés, por cada elemento de ésta, se obtiene el último nodo de su ruta (columna seleccionada por el usuario) y se incluye como un elemento (columna) de la cláusula *Select*.

En la Figura 4.10 se muestra un ejemplo de la construcción de una cláusula *Select* con tres rutas, cada una de éstas pertenece a un elemento de interés especificado por un usuario. Para tal efecto, se toma el último nodo de cada ruta y se inserta en la cláusula *Select* la columna referente al nodo.

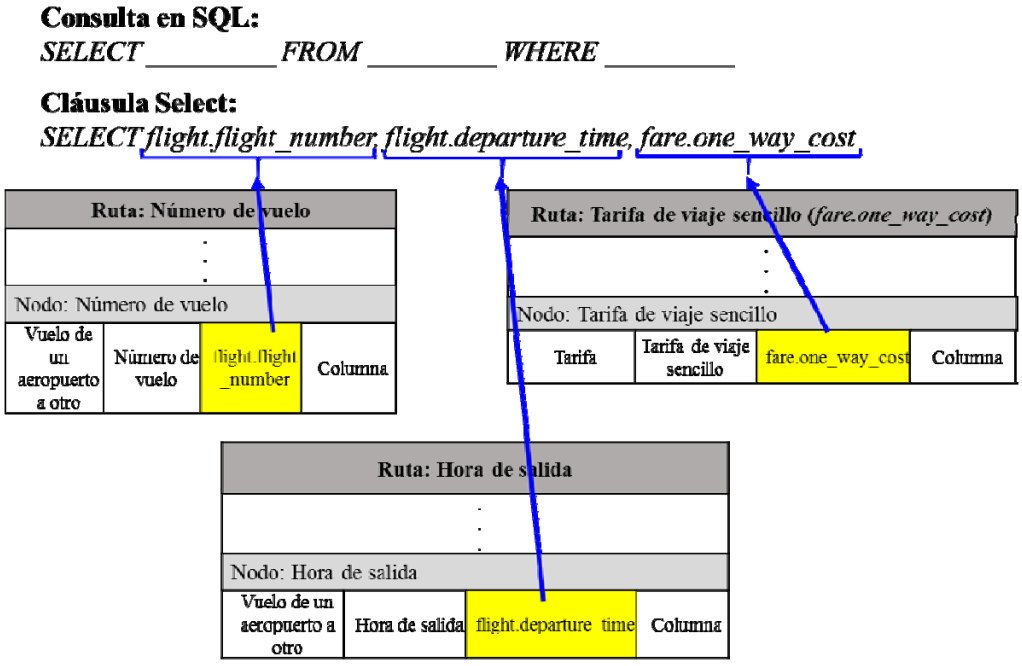


Figura 4.10. Construcción de la cláusula Select.

Para la cláusula *Where*, a partir de la lista de condiciones de búsqueda, por cada condición especificada por el usuario, se toman el nombre de la columna, el operador de comparación y el valor de la condición para generar una restricción en la cláusula, tal como se ilustra en la Figura 4.11. Cabe señalar que la interfaz ha sido diseñada para especificar restricciones que se apliquen en conjunto, es decir, que todas las condiciones se cumplan (elementos asociados por la palabra AND en la cláusula *Where*).

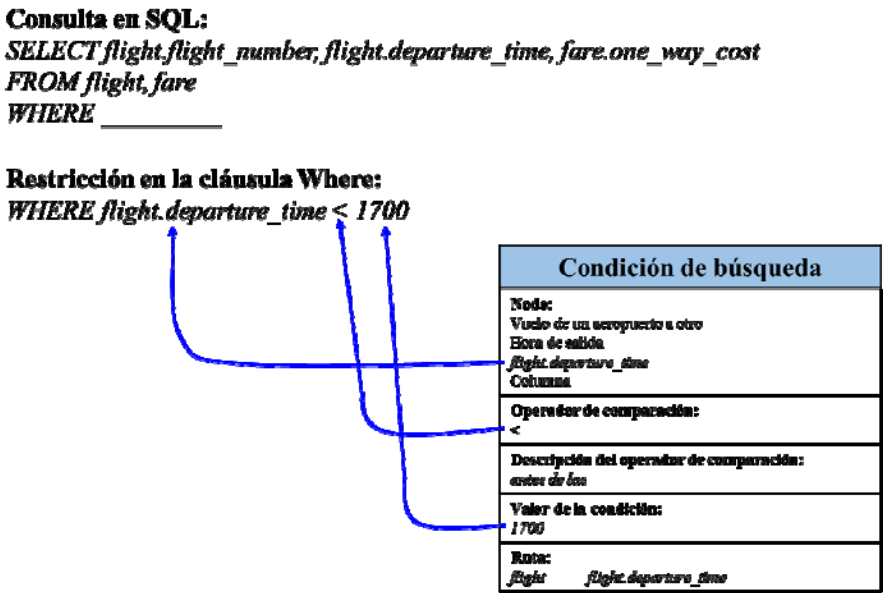


Figura 4.11. Construcción de restricciones en la cláusula Where

Posteriormente, se construyen las rutas que asocian los elementos de interés y condiciones de búsqueda con la tabla principal seleccionada al comienzo del proceso de composición para que la consulta tenga coherencia. Lo anterior se realiza definiendo las reuniones entre las tablas involucradas a partir de las rutas definidas en la lista de elementos de interés y las rutas de la lista de condiciones de búsqueda, donde cada elemento de una ruta implica una reunión. Por lo tanto, por todas las rutas de los elementos de interés se toman las reuniones especificadas en las rutas de cada elemento y de igual manera para las condiciones de búsqueda, sin repetir reuniones ya encontradas. El proceso descrito se puede observar en la Figura 4.11.

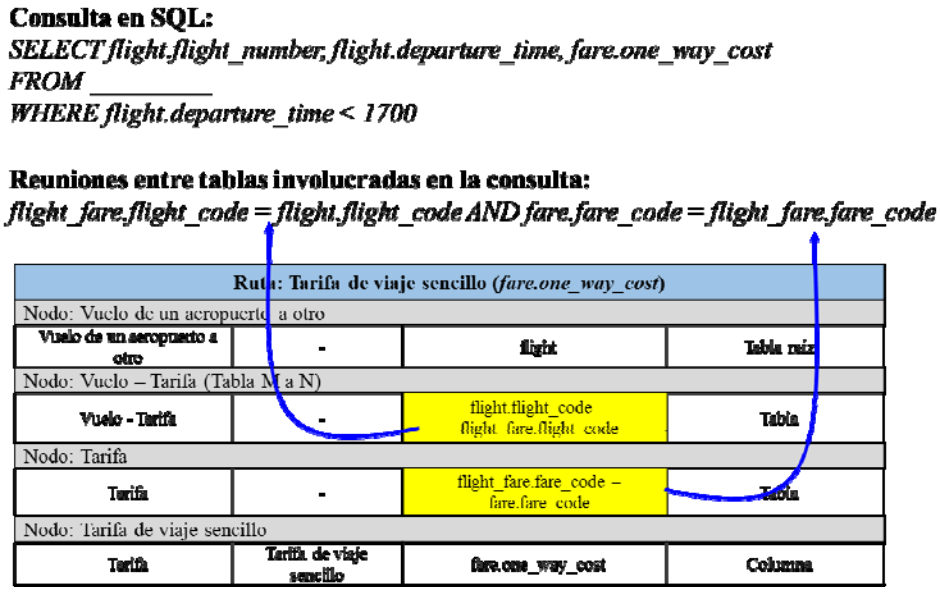


Figura 4.11. Construcción de las reuniones entre tablas en la cláusula Where

Por último se construye la cláusula *From* de la consulta a partir de las tablas encontradas en las rutas de los elementos de interés y las rutas de las condiciones de búsqueda. Este proceso se efectúa como se ilustra en la Figura 4.12, donde se revisan uno a uno los nodos de las rutas y se almacenan sin repetir tablas ya almacenadas.

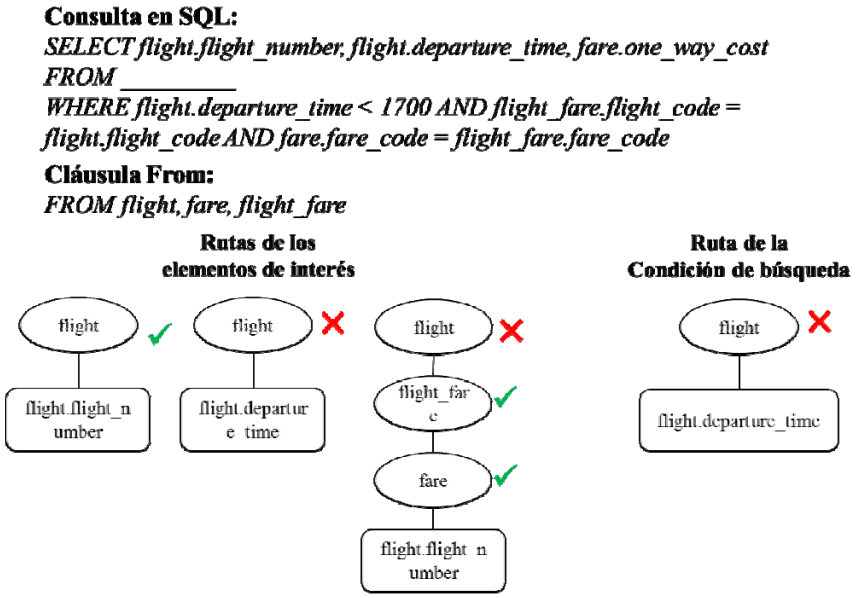


Figura 4.12. Construcción de la cláusula From

4.6 Vista previa y resultados de la consulta

Durante el proceso de composición de una consulta, el usuario define los elementos de las cláusulas *Select* y *Where* de la consulta mediante la interfaz de composición. Posteriormente la interfaz

compone la consulta a partir de dichos elementos y procesa dicha consulta a través del sistema manejador de base de datos (SMBD) para obtener el resultado de la consulta.

En la ventana de *Vista previa y resultado de la consulta* (ver la Figura 4.13), se muestran las selecciones que realizó el usuario en las ventanas anteriores; además, también se incluye la consulta compuesta en SQL así como su resultado en forma de tabla. Dicha tabla contiene la información que desea conocer el usuario.

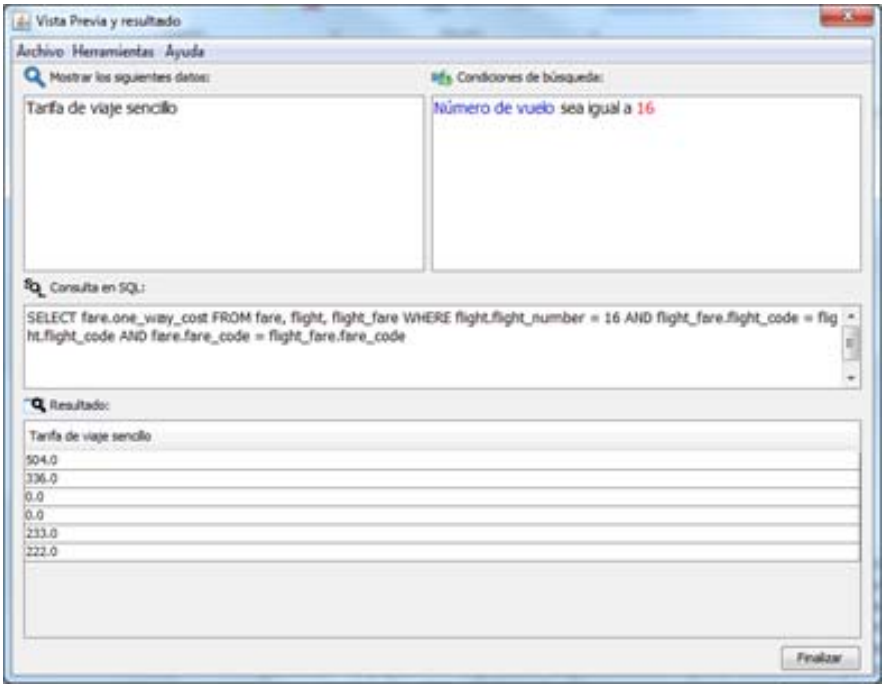


Figura 4.13. Vista previa y resultado de la consulta

4.7 Proceso de composición de consultas

Para describir el proceso de composición se considera la siguiente consulta:

Muestra la tarifa de viaje sencillo del vuelo número 16.

El primer paso consiste en definir el tema de interés (tabla principal) de la consulta. La interfaz muestra una lista con las descripciones de todos los temas (tablas) disponibles en la base de datos agrupadas por nivel de relevancia. Para este ejemplo, el usuario debe seleccionar V*uelo de un aeropuerto a otro* (tabla *flight*) como tema de interés, así como se muestra en la Figura 4.14.

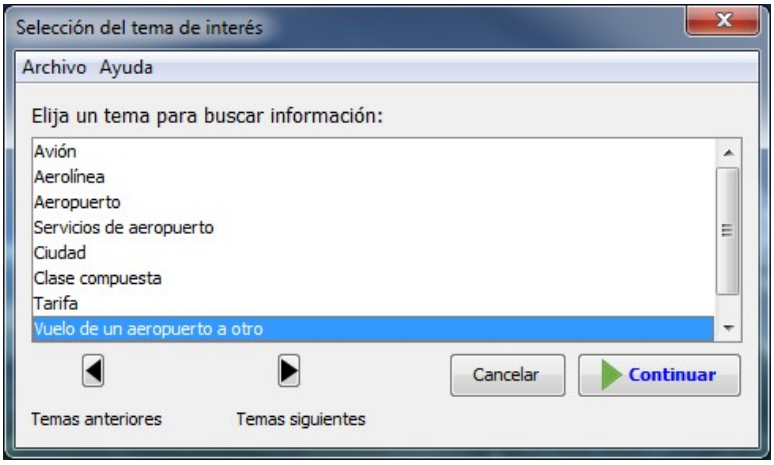


Figura 4.14. Selección del tema de interés

En el segundo paso, la interfaz construye el árbol de composición a partir del tema seleccionado y la información obtenida del SID (relaciones entre cada tabla de la BD y la tabla raíz) y muestra una ventana donde el usuario puede elegir los elementos del tema principal o elementos de los temas relacionados cuya información sea de interés. Este paso de la composición es equivalente a definir la cláusula Select de una consulta en SQL.

Como se muestra en la consulta de ejemplo, el usuario desea conocer la *tarifa de viaje sencillo* de un vuelo. Este elemento se encuentra en un tema relacionado al tema principal, específicamente dentro del tema relacionado *Tarifa* (tabla *fare*): el elemento llamado *Tarifa de viaje sencillo* (columna *one_way_fare*). Por lo tanto, el usuario debe seleccionar *Tarifa de viaje sencillo* del árbol de composición y añadirlo a la lista de elementos a mostrar como se muestra en la figura 4.15.

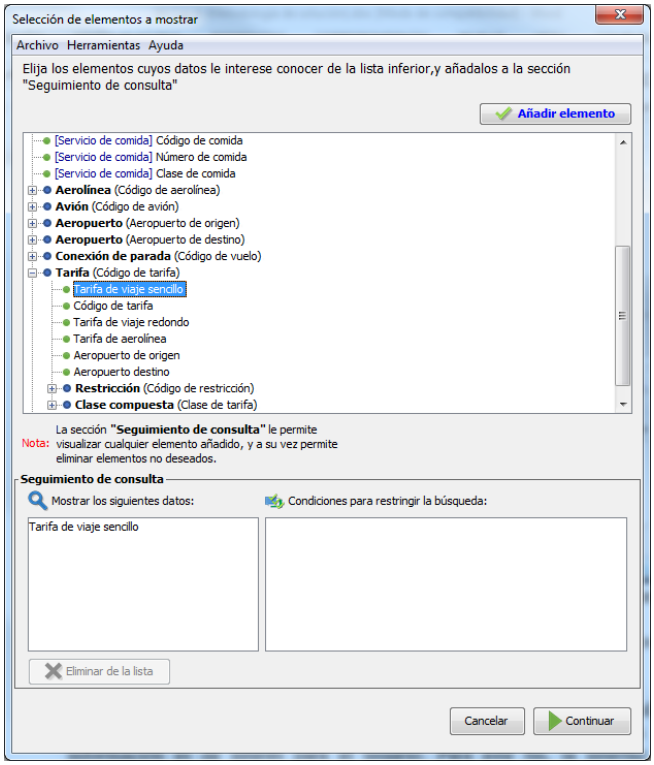


Figura 4.15. Selección de los elementos de los temas

Una vez que el usuario haya terminado de añadir tantos elementos a la lista como lo desee, la interfaz preguntará si éste desea definir condiciones de búsqueda para restringir la información a obtener, entonces el usuario indica sí desea especificar condiciones de búsuqueda.

El último paso consiste en especificar las condiciones de búsqueda de la consulta; en otras palabras, las condiciones para restringir el conjunto de todos los posibles elementos (entidades) cuya información es de interés para el usuario. Para este fin, la interfaz muestra el mismo árbol de composición, excepto que esta vez el usuario debe seleccionar uno o más elementos que servirán para restringir el conjunto de elementos/entidades; además, por cada elemento de restricción, el usuario debe especificar una condición, la cual está compuesta de un valor y un operador de comparación (*igual a, menor que, menor o igual que*, etc.).

Para la consulta de ejemplo, el usuario desea conocer la *tarifa de viaje sencillo del vuelo número 16*; en este caso el usuario está interesado en conocer la tarifa de un conjunto de vuelos cuyo *número de vuelo* sea *16.* Por lo tanto, el usuario debe seleccionar el elemento *Número de vuelo* (columna *flight.flight_number*) situado en el tema *Vuelo de un aeropuerto a otro* del árbol de composición y especificar que *sea igual a 16* mediante la lista desplegable que contiene las descripciones de los operadores de comparación y la caja de texto para introducir un valor, como se muestra en la Figura 4.16. Finalmente, el usuario debe añadir esta condición a la lista de condiciones de búsqueda. Este paso es equivalente a definir la cláusula Where de una consulta en SQL.

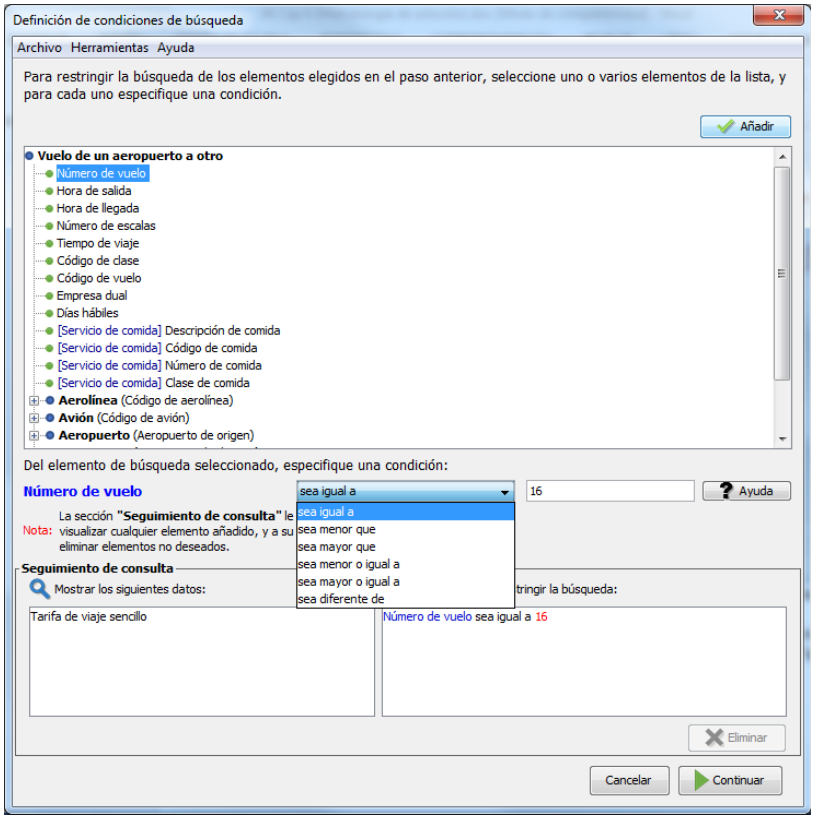


Figura 4.16. Especificación de las condiciones de búsqueda.

Capítulo 5

Interfaz de configuración

Para que el usuario pueda hacer uso de la interfaz de composición, es necesario efectuar una configuración de la misma, la cual se lleva a cabo por el administrador de la BD mediante la interfaz de configuración. Dicha interfaz permite al administrador de la BD modificar el contenido del DIS. La información relevante para la composición de consultas que puede ser modificada mediante la interfaz de configuración se muestra en la Tabla 5.1. Es importante mencionar que el DIS forma parte del trabajo presentado en [Aguirre, 2014], donde éste se genera a partir de la BD. Por consiguiente, algunos elementos (nombres de tablas, nombres de columnas y sus conexiones) no podrán ser modificados por la interfaz de configuración perteneciente a este trabajo.

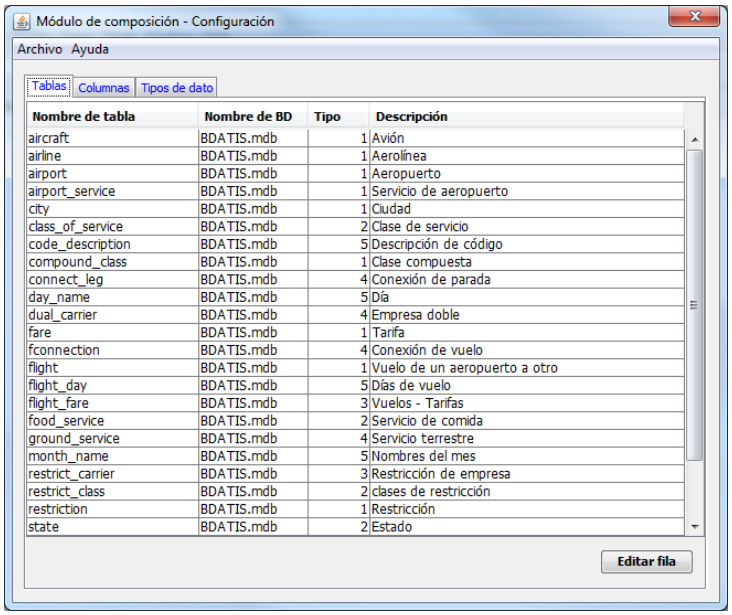
Tabla 5.1. Información modificable mediante la interfaz de configuración

Por cada tabla en la BD	
Tipo	El tipo de tabla al que pertenece. Puede pertenecer sólo a un tipo de los siguientes: 1. Tabla base. 2. Catálogo. 3. Relación M a N. 4. Tabla satélite.
Descripción	Descripción de la tabla que se mostrará en el árbol de composición.
Por cada columna en la BD	
Orden	Número que sirve para identificar el orden en el cual se mostrarán las columnas de una tabla en el árbol de composición, siendo las columnas con número menor las primeras en mostrarse.
Tipo de dato	Tipo de dato asociado a la columna. Solamente se pueden asignar tipos de dato creados por el administrador de la BD mediante la interfaz de configuración.
Descripción	Descripción de la columna que se mostrará en el árbol de composición.
Texto de ayuda	Texto que describe la finalidad de una columna en la BD; además, contiene ejemplos sobre el formato que tiene dicha columna.
Por cada tipo de dato personalizado	
Operadores de comparación compatibles	Conjunto de operadores de comparación que pueden ser utilizados en conjunción con un tipo de dato personalizado en el proceso de composición de consultas (=, <, >, <=, >=, o <>).
Descripción de operadores de comparación	Texto que describe la función de cada operador de comparación compatible con el tipo de dato personalizado; p. ej., para expresar el operador > referente a un tipo de dato hora, puede definirse

como *después de las.*

5.1 Configuración de tablas

La interfaz de configuración consta de una ventana principal con tres pestañas, las cuales sirven para configurar los aspectos referentes a las tablas, columnas y tipos de datos. Como se puede observar en la Figura 5.1, la pestaña *Tablas* de la interfaz permite al administrador de la BD visualizar el nombre de todas las tablas contenidas en la BD, así como su tipo y descripción. Dichas tablas se muestran ordenadas en orden alfabético y sólo se permite editar los elementos de una tabla a la vez.



Nombre de tabla	Nombre de BD	Tipo	Descripción
aircraft	BDATIS.mdb	1	Avión
airline	BDATIS.mdb	1	Aerolínea
airport	BDATIS.mdb	1	Aeropuerto
airport_service	BDATIS.mdb	1	Servicio de aeropuerto
city	BDATIS.mdb	1	Ciudad
class_of_service	BDATIS.mdb	2	Clase de servicio
code_description	BDATIS.mdb	5	Descripción de código
compound_class	BDATIS.mdb	1	Clase compuesta
connect_leg	BDATIS.mdb	4	Conexión de parada
day_name	BDATIS.mdb	5	Día
dual_carrier	BDATIS.mdb	4	Empresa doble
fare	BDATIS.mdb	1	Tarifa
fconnection	BDATIS.mdb	4	Conexión de vuelo
flight	BDATIS.mdb	1	Vuelo de un aeropuerto a otro
flight_day	BDATIS.mdb	5	Días de vuelo
flight_fare	BDATIS.mdb	3	Vuelos - Tarifas
food_service	BDATIS.mdb	2	Servicio de comida
ground_service	BDATIS.mdb	4	Servicio terrestre
month_name	BDATIS.mdb	5	Nombres del mes
restrict_carrier	BDATIS.mdb	3	Restricción de empresa
restrict_class	BDATIS.mdb	2	clases de restricción
restriction	BDATIS.mdb	1	Restricción
state	BDATIS.mdb	2	Estado

Figura 5.1. Interfaz de configuración (tablas)

La modificación del tipo o descripción de una tabla, puede realizarse seleccionando una fila y haciendo clic sobre el botón *Editar fila* o en su defecto haciendo doble clic sobre la fila de la tabla que desea modificar. Para realizar las modificaciones, la interfaz muestra una ventana emergente (Figura 5.2) que contiene los elementos que pueden ser modificados, los cuales son: el tipo (lista) y la descripción (caja de texto). Como se puede observar, el tipo de tabla puede ser seleccionado a partir de una lista que contiene los tipos de tabla definidos en el núcleo de la interfaz. Es necesario tener en cuenta que para realizar una configuración correcta se aconseja ver la Tabla 3.1. Asimismo, el administrador puede definir la descripción que desee que se muestre para la tabla.

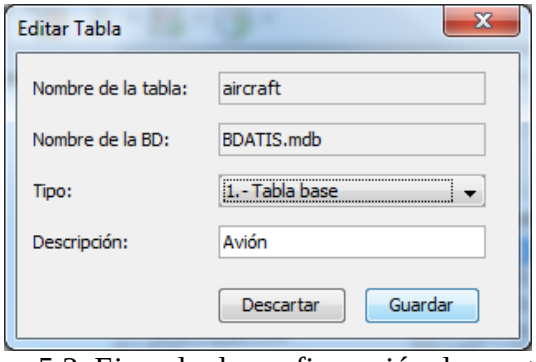
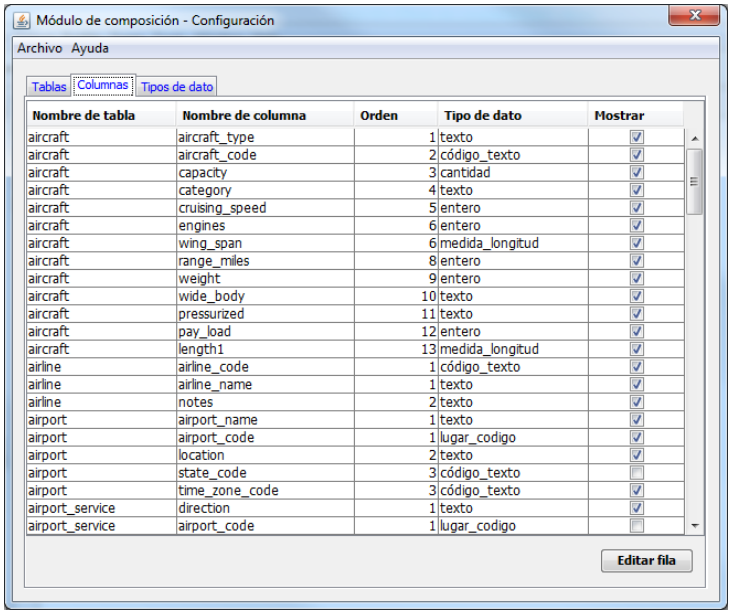


Figura 5.2. Ejemplo de configuración de una tabla

Una vez que se hayan realizado las modificaciones correspondientes, hacer clic en el botón *Guardar* para almacenar los cambios.

5.2 Configuración de columnas

La configuración de las columnas de la BD se realiza en la pestaña *Columnas* (Figura 5.3), en la cual se muestran el nombre de tabla, nombre de columna, orden, tipo de dato y si se muestra o no dicha columna en el árbol de composición. Los elementos mencionados se muestran en la tabla para todas las columnas de la BD. De la misma manera que en la pestaña de *Tablas*, la modificación se realiza seleccionando una fila y haciendo clic en el botón *Editar fila* o haciendo doble clic en la fila de la columna a modificar.



Nombre de tabla	Nombre de columna	Orden	Tipo de dato	Mostrar
aircraft	aircraft_type	1	texto	☑
aircraft	aircraft_code	2	código_texto	☑
aircraft	capacity	3	cantidad	☑
aircraft	category	4	texto	☑
aircraft	cruising_speed	5	entero	☑
aircraft	engines	6	entero	☑
aircraft	wing_span	6	medida_longitud	☑
aircraft	range_miles	8	entero	☑
aircraft	weight	9	entero	☑
aircraft	wide_body	10	texto	☑
aircraft	pressurized	11	texto	☑
aircraft	pay_load	12	entero	☑
aircraft	length1	13	medida_longitud	☑
airline	airline_code	1	código_texto	☑
airline	airline_name	1	texto	☑
airline	notes	2	texto	☑
airport	airport_name	1	texto	☑
airport	airport_code	1	lugar_codigo	☑
airport	location	2	texto	☑
airport	state_code	3	código_texto	☐
airport	time_zone_code	3	código_texto	☑
airport_service	direction	1	texto	☑
airport_service	airport_code	1	lugar_codigo	☐

Figura 5.3. Interfaz de configuración (columnas)

Para modificar los elementos de una columna, la interfaz muestra una ventana emergente (Figura 5.4) que contiene los siguientes elementos:

- Nombre de la tabla. Nombre de la tabla a la cual pertenece la columna. Este elemento no puede ser modificado.
- Nombre de la columna. Nombre de una columna. Este elemento no puede ser modificado.
- Orden. Número que indica en qué orden se presenta una columna en el árbol de composición. Para modificar este elemento se muestra una lista con números a partir del 1 hasta *n*, donde *n* es el número total de columnas de la tabla a la que pertenece una columna. Las columnas se muestran en el árbol de composición en orden ascendente de acuerdo a este número.
- Tipo de dato. Tipo de dato asociado una columna, el cual ha sido creado por el administrador de BD, y puede ser seleccionado mediante una lista que contiene todos los tipos de dato personalizados disponibles en el DIS.
- Mostrar. Indica si una columna se muestra o no en el árbol de composición.

- Descripción. Texto que se muestra en el árbol de composición para hacer referencia a una columna. Puede ser modificado mediante la caja de texto.
- Texto de ayuda. Texto que describe la finalidad de una columna en la BD. Además contiene ejemplos sobre el formato que tienen los datos de dicha columna. Puede ser modificado mediante la caja de texto designada y además, el texto podrá contener etiquetas en código HTML para dar formato al texto.

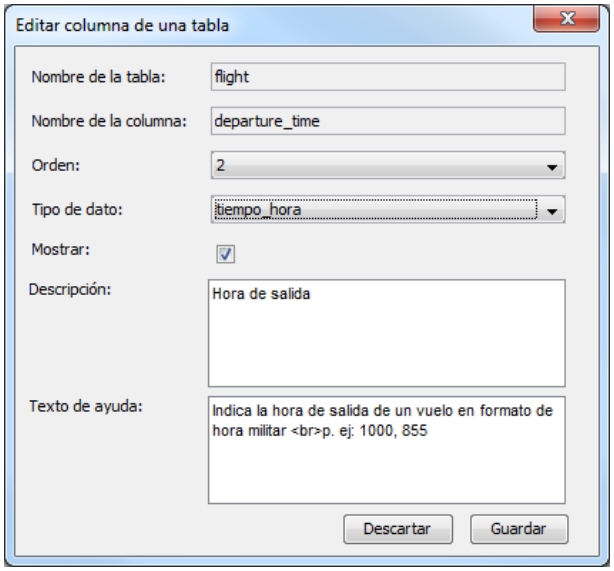


Figura 5.4. Ejemplo de configuración de una columna

Una vez que se hayan realizado las modificaciones correspondientes, hacer clic en el botón *Guardar* para almacenar los cambios.

5.3 Tipos de dato personalizados

Por último, la información correspondiente a los tipos de dato puede ser vista y modificada en la pestaña *Tipos de dato* (Figura 5.5), la cual contiene una lista de todos los tipos de dato que han sido creados por el administrador de la BD.

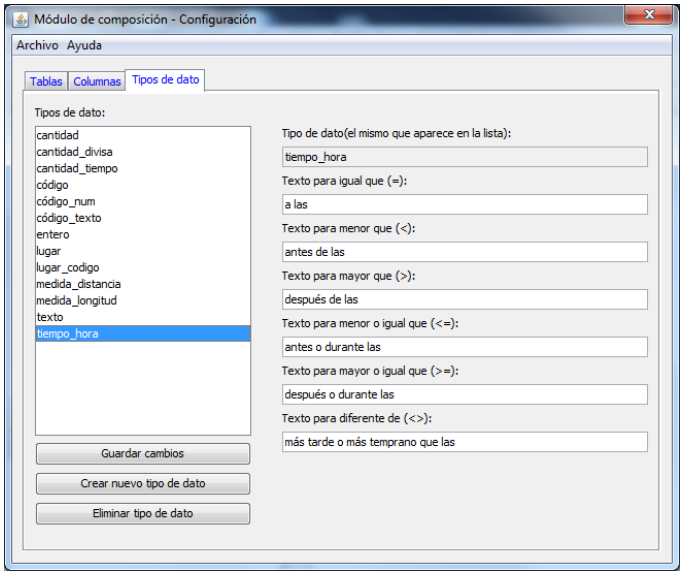


Figura 5.5. Interfaz de configuración (tipos de dato)

Un tipo de dato tiene asociado seis descripciones, una por cada operador de comparación (=, <, >, <=. >=, <>) que pueda ser utilizado para componer consultas.

Para modificar las descripciones de los operadores de comparación, es necesario seleccionar un tipo dato de la lista y posteriormente modificar la descripción deseada que aparece en la caja de texto correspondiente. Para que las modificaciones tomen efecto se debe hacer clic en el botón *Guardar cambios*.

Además, para eliminar un tipo de dato se debe seleccionar un tipo de dato y hacer clic en el botón *Eliminar tipo de dato*; asimismo, para crear un tipo de dato se debe hacer clic en el botón *Crear nuevo tipo de dato* y la interfaz mostrará una ventana emergente donde se especificarán las características del nuevo tipo de dato. Dicha ventana se muestra en la Figura 5.6.

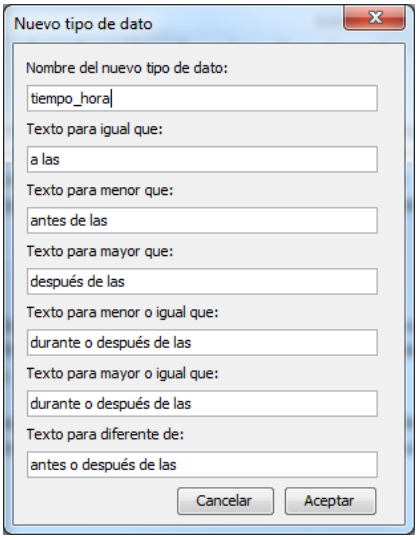


Figura 5.6. Ejemplo de creación de un tipo de dato

Como se puede observar en la Figura 5.6, se define una descripción por cada operador de comparación, dejando en blanco la descripción correspondiente al operador que no se desee mostrar en el proceso de composición.

Una vez que el administrador de BD termine de especificar las descripciones de cada operador de comparación, se procede a hacer clic en *Aceptar* para terminar la creación del tipo de dato.

Capítulo 6

Experimentación

6.1 Pruebas sobre consultas del tipo 1 al 4

En esta subsección se presentan las pruebas realizadas a la interfaz de composición. Dichas pruebas cubren las consultas de tipo 1 al 4 mencionados en la subsección 3.3. Para tal efecto, las pruebas fueron realizadas por estudiantes de ingeniería en ciencias de la computación.

6.1.1 Objetivo de las pruebas

Las pruebas realizadas a la interfaz de composición tienen como fin medir la facilidad de uso de ésta en conjunción con la funcionalidad de la misma. Esto se hace para saber si la interfaz es suficientemente amigable y funcional para poder componer consultas del tipo 1 al 4 (véase Capítulo 3) sobre una base de datos compleja. Para tal efecto se midieron los siguientes parámetros:

- Cantidad de tiempo que le toma a un usuario componer una consulta.
- Cantidad de intentos que realiza un usuario para componer una consulta correctamente.

Considerando los parámetros mencionados, se puede obtener información sobre la dificultad que implica la composición de consultas de un determinado tipo.

6.1.2 Descripción del ambiente de pruebas

Las pruebas se llevaron a cabo en una sesión de una hora con 17 estudiantes de ingeniería en ciencias de la computación, a los cuales se les proporcionó con un día de anticipación un manual de usuario que detalla la función de los controles de la interfaz y ejemplos de composición de cuatro consultas sencillas. Al comenzar la prueba se proporcionó a los usuarios un conjunto de 20 consultas en

lenguaje natural ordenadas por nivel de dificultad, las cuales se encuentran separadas por grupos de 5 consultas, ordenadas de la siguiente manera:

- Consultas de la 1 a la 5. Consultas pertenecientes al tipo 1 (involucran 1 tabla base).
- Consultas de la 6 a la 10. Consultas pertenecientes al tipo 2 (involucran 2 tablas base).
- Consultas de la 11 a la 15. Consultas pertenecientes al tipo 3 (involucran 3 tablas base).
- Consultas de la 16 a la 20. Consultas pertenecientes al tipo 4 (involucran 2 tablas base y una tabla de relación M a N).

Dichas consultas fueron diseñadas de tal forma que ninguna presentara problemas de elipsis semántica, es decir, que la consulta no omitiera palabras que pudieran ser cruciales para la interpretación de la misma. Por ejemplo, la consulta *Dame la tarifa del vuelo 19* se reformuló de la siguiente manera *Dame la tarifa de viaje sencillo (o redondo) del vuelo cuyo número sea 19*. Esto se necesita debido a que los usuarios no son expertos en el dominio de la BD (vuelos y aerolíneas), por lo tanto, las consultas fueron diseñadas de la manera más explícita posible. En este punto es importante mencionar que **a los estudiantes nunca se les proporcionó el esquema de la base de datos (Apéndice A) ni las consultas correctas en SQL**.

Para verificar y recolectar los resultados obtenidos de las consultas compuestas por los usuarios, se implementó una versión de la interfaz de composición especial para realizar las pruebas. El funcionamiento de dicha implementación se puede observar en la Figura 6.1.

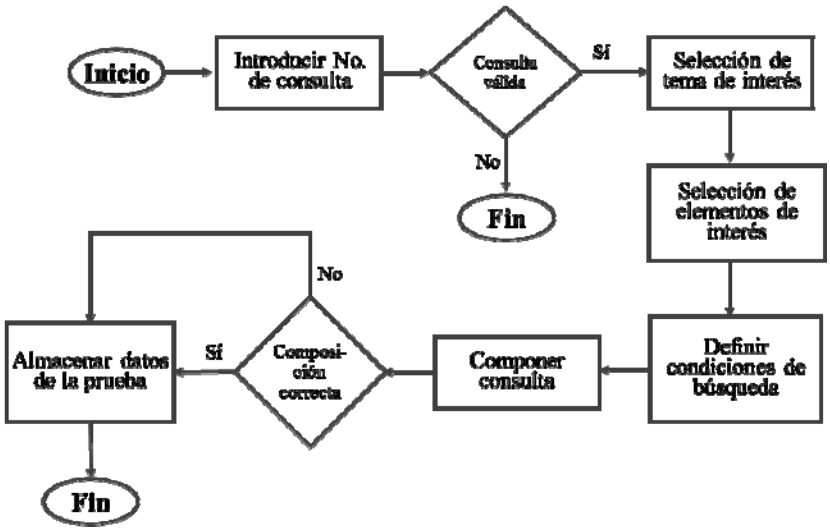


Figura 6.1. Funcionamiento de la interfaz para pruebas

Por cada consulta que el usuario intente componer, deberá introducir el número de consulta correspondiente. Enseguida, la interfaz determina si el número de consulta es válido; en caso afirmativo, la interfaz procede a mostrar las ventanas correspondientes al proceso de composición. Después, cuando la interfaz termina de generar la expresión en SQL basándose en las selecciones del usuario, ésta determina si la consulta en SQL concuerda con la consulta correcta que se encuentra almacenada; entonces se almacena la consulta compuesta, el identificador de usuario efectuó la composición, el tiempo de composición, y si la consulta compuesta concuerda con la correcta o no.

Una vez que el usuario ha compuesto una consulta correctamente, no puede volver a componer la misma consulta. Además, la interfaz de prueba permite intentar componer una consulta tantas veces como sea necesario hasta que ésta sea compuesta de manera correcta.

6.1.3 Resultados

Los resultados obtenidos de las pruebas se muestran en la Tabla 6.1. Dichos resultados se obtienen a partir de 20 consultas compuestas por 17 usuarios, descartando los intentos de las consultas que los usuarios no hayan podido componer correctamente.

Tabla 6.1. Resultados por consulta

No. consulta	Mínimo de intentos	Máximo de intentos	Promedio de intentos	Mínimo de tiempo (seg.)	Máximo de tiempo (seg.)	Tiempo promedio (min.)	No. de usuarios comp. correcta
1	1	10	2.35	42.00	609.00	2.78	17
2	1	6	1.47	43.00	343.00	1.55	17
3	1	3	1.25	27.00	303.00	1.49	16
4	1	2	1.12	31.00	73.00	0.76	17
5	1	7	1.59	28.00	285.00	1.17	17
6	1	2	1.06	24.00	105.00	0.71	17
7	1	1	1.00	20.00	85.00	0.59	16
8	1	5	2.00	24.00	280.00	1.36	17
9	1	1	1.00	27.00	57.00	0.59	17
10	1	6	2.15	41.00	514.00	2.55	13
11	1	1	1.00	34.00	97.00	0.83	17
12	1	2	1.06	39.00	123.00	0.98	16
13	1	2	1.06	46.00	182.00	1.23	16
14	1	4	1.24	24.00	185.00	0.91	17
15	1	4	1.18	55.00	257.00	1.63	17
16	1	1	1.00	19.00	75.00	0.49	17
17	1	2	1.06	31.00	99.00	0.82	17
18	1	2	1.24	26.00	110.00	0.84	17
19	1	3	1.25	25.00	237.00	1.30	16
20	1	8	4.64	90.00	970.00	6.93	11
Promedio			**1.48**			**1.47**	

A continuación se explica el significado de las columnas de la Tabla 6.1. La primera columna indica el número de consulta, la segunda columna indica el número mínimo de intentos que le tomó a un usuario componer correctamente la consulta, la tercera columna indica el número máximo de intentos, la cuarta columna muestra el número promedio de intentos, la quinta columna indica el tiempo mínimo en segundos que empleó un usuario para componer la consulta, la sexta columna indica el tiempo máximo en segundos, la séptima columna muestra el tiempo promedio en minutos, por último, la octava columna indica el número de usuarios que lograron componer correctamente la consulta.

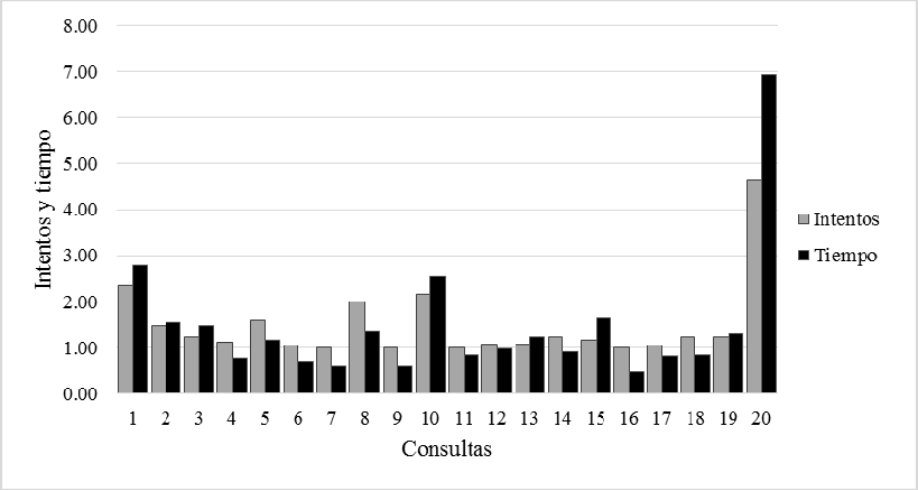


Figura 6.2. Promedio de intentos y tiempo por consulta

De la tabla 6.1, es importante destacar los siguientes aspectos:

- la mayoría de los usuarios compusieron las consultas de la 4 a la 19 a excepción de la 10 en el primer intento, y el promedio de tiempo máximo para ese conjunto de consultas es de 150 segundos.
- En todas las consultas al menos un usuario compuso la consulta correspondiente en un intento; esto se ve reflejado en la tabla 6.1, en la columna *Mínimo de intentos* que indica un intento como mínimo por cada consulta.
- El número máximo de intentos de todas las consultas se observa en la consulta 1, la cual como se mencionó, permite al usuario comenzar a aprender a usar la interfaz.
- Al igual que en los intentos, los tiempos máximos más altos se presentan en las consultas 1 y 10, dichos tiempos en comparación con los tiempos máximos de las demás consultas los duplican o triplican; indicando que las consultas 1 y 10 son casos aislados y los tiempos elevados no se deben al diseño de la interfaz.
- El promedio de usuarios que compusieron correctamente todas las consultas es de 16.

Como se puede observar en la gráfica de la Figura 4, la mayoría de las consultas pudieron ser respondidas por los usuarios en 1 o 2 intentos y en un tiempo no mayor a 2 minutos. Sin embargo, en las consulta 1 se presenta un mayor promedio de intentos y tiempo. Esto se debe a que ésta es la primera consulta que deben componer y en ésta se enfrentan al proceso de aprendizaje de la operación de la interfaz. Posteriormente los intentos y el tiempo promedio van disminuyendo debido a que los usuarios se acoplan al funcionamiento de la interfaz. Es importante destacar que para la cuarta consulta los usuarios ya aprendieron a operar la interfaz.

La consulta 10 (*Dame los códigos de clase de tarifa y tipos de tarifa de temporada de la clase de servicio con rango 12*) requirió un número de intentos mayor debido a que la consulta requiere información más específica y requiere que el usuario tenga un mayor conocimiento sobre el tema a tratar.

Además la consulta 20 (*Dame los números de vuelo y hora de llegada de los vuelos que salen desde San Francisco y llegan antes de las 2100 hrs.*) requirió un número elevado de intentos debido a

que involucra un gran número de tablas, y la información requerida por la consulta necesita ser buscada en los niveles más profundos del árbol de composición.

6.1.4 Conclusiones

Las pruebas efectuadas a la interfaz de composición demuestran que la interfaz sólo requiere que el usuario utilice la interfaz un mínimo número de veces para componer consultas más rápida y eficientemente. Esto se debe a que la interfaz es lo suficientemente intuitiva para que un usuario que desconoce el esquema de BD y el tema de la BD pueda componer consultas en 1 minuto aproximadamente.

Cabe señalar que las consultas en lenguaje natural propuestas en estas pruebas fueron diseñadas por una tercera persona, por lo que la interfaz tendría un mejor desempeño si un usuario diseñara sus propias consultas, debido a que éste conoce específicamente los datos que desea obtener de la BD.

Los resultados mostrados en la tabla 6.1 indican que de un total de 20 consultas, 13 pudieron ser compuestas correctamente por todos los usuarios. Además, otras 5 consultas fueron compuestas correctamente por 16 usuarios, dejando 2 consultas con 13 y 11 composiciones correctas respectivamente. Esto demuestra que la mayoría de las consultas pudieron ser compuestas correctamente por los usuarios. Aquellas consultas cuyo número de composiciones correctas fueron bajos (consultas 10 y 20), se debe a que los usuarios con los que se efectuaron las pruebas no están familiarizados con el dominio de la BD.

Es importante mencionar que 11 de 17 usuarios compusieron correctamente todas las consultas. Esto demuestra que la interfaz puede ser usada para componer consultas por la mayoría de los usuarios; lo anterior es considerando que el esquema de la base de datos nunca se proporcionó a los usuarios involucrados en las pruebas.

Las pruebas efectuadas a la interfaz de composición demuestran que ésta sólo requiere que el usuario utilice la interfaz un mínimo número de veces (aproximadamente tres) para poder componer consultas rápida y eficientemente. Esto es debido a que la interfaz es lo suficientemente intuitiva como para que un usuario que desconoce el esquema de BD y el tema de la BD pueda componer consultas en un minuto aproximadamente.

6.2 Pruebas funcionales sobre consultas del tipo 5

Para demostrar que la interfaz propuesta permite componer consultas del tipo 5 (consultas que involucren tres tablas base conectadas mediante dos tablas de relación M a N), se realizan las pruebas de funcionalidad descritas en las siguientes subsecciones.

6.2.1 Descripción de las pruebas

Las pruebas descritas en esta subsección fueron realizadas con el fin de demostrar que es posible componer consultas del tipo 5 (consultas que involucren tres tablas base conectadas mediante dos tablas que implementen una relación M a N) haciendo uso de la interfaz de composición.

Para tal efecto, se diseñaron tres consultas de dicho tipo y se realizó su composición mediante la interfaz de composición. A continuación se describen los detalles correspondientes a la composición de dichas consultas.

Consulta 1:

Dame el código de conexión, el código de aeropuerto de origen y destino, y la tarifa de viaje sencillo del vuelo número 314.

La consulta número 1 plantea que el usuario desea conocer información sobre las siguientes columnas de la base de datos:

- Código de conexión (*fconnection.connect_code*).
- Código de aeropuerto de origen (*flight.from_airport*).
- Código de aeropuerto de destino (*flight.to_ariport*).
- Tarifa de viaje sencillo (*fare.one_way_cost*).

Además, dicha consulta plantea la definición de la siguiente condición de búsqueda:

- Número de vuelo sea igual a 314 (*flight.flight_number = 314*).

Como se puede observar, la consulta anterior requiere información de las siguientes tablas: *fconnection*, *flight* y *fare*. Además, la especificación de la condición de búsqueda ya mencionada se centra en la tabla *flight*. Considerando las columnas mencionadas, la interfaz genera las reuniones entre las tablas correspondientes:

fconnection.connect_code = connect_leg.connect_code
connect_leg.flight_code = flight.flight_code
flight_fare.flight_code = flight.flight_code
fare.fare_code = flight_fare.fare_code

Considerando las reuniones generadas por la interfaz, se observa que las tablas involucradas en la consulta son tres tablas base y dos tablas que implementan una relación M a N: *fconnection* (tabla base), *connect_leg* (tabla M a N), *flight* (tabla base), *flight_fare* (tabla M a N), *fare* (tabla base).

Consulta 2:

Dame el número de vuelo y la tarifa de viaje redondo de los vuelos que salen de la ciudad de San Francisco.

Esta consulta implica la obtención de la siguiente información, número de vuelo y tarifa de viaje redondo. Dicha información se encuentra en las columnas *flight.flight_number* y *fare.rnd_trip_cost.*

La condición de búsqueda para esta consulta es la siguiente: que la ciudad de salida del vuelo sea San Francisco. Dicha condición se representa de la siguiente manera: *city.city_name = 'San Francisco'*.

La interfaz de composición genera reuniones entre las tablas *fare*, *flight* y *city*. Las reuniones quedan generadas de la siguiente manera:

flight_fare.flight_code = flight.flight_code
fare.fare_code = flight_fare.fare_code
airport.airport_code = flight.from_airport
airport_service.airport_code = airport.airport_code
city.city_code = airport_service.city_code

Se puede apreciar que las tablas involucradas en la consulta son cuatro tablas base y dos tablas que implementan una relación M a N. Dichas tablas son las siguientes: *fare* (tabla base), *flight_fare* (tabla M a N), *flight* (tabla base), *airport* (tabla base), *airport_service* (tabla M a N), *city* (tabla base).

Consulta 3:

Dame el código de conexión y el número de vuelo de los vuelos que salen de Atlanta.

La información que se desea conocer en esta consulta es el código de conexión (*fconnection.connect_code*) y número de vuelo (*flight.flight_number*). La condición de búsqueda para dicha consulta hace referencia a la columna *city.city_name = 'Atlanta'*.

Las reuniones generadas por la interfaz de composición son las siguientes:

fconnection.connect_code = connect_leg.connect_code
connect_leg.flight_code = flight.flight_code
airport.airport_code = flight.from_airport
airport_service.airport_code = airport.airport_code
city.city_code = airport_service.city_code

Las tablas involucradas en la consulta son cuatro tablas base y dos tablas que implementan una relación M a N. Dichas tablas son las siguientes: *fconnection* (tabla base), *connect_leg* (tabla M a N), *flight* (tabla base), *airport* (tabla base), *airport_service* (tabla M a N), *city* (tabla base).

Las tres consultas descritas en esta subsección pueden ser compuestas mediante la interfaz de composición de consultas.

Para tal efecto, el usuario debe hacer uso del árbol de composición para especificar los elementos que desea conocer de la BD y las condiciones de búsqueda para dichos elementos. Es importante mencionar que el árbol de composición puede expandirse hasta tres niveles por cada relación existente con la tabla raíz. A pesar de dicha limitación, la interfaz de composición es capaz de componer consultas del tipo 5, las cuales necesitan que el usuario explore el árbol de composición hasta sus niveles más profundos.

Capítulo 7

Conclusiones y trabajos futuros

En este capítulo se presentan las conclusiones y aportaciones que ha realizado este trabajo, así como algunos trabajos futuros que podrían aumentar la funcionalidad de la interfaz desarrollada y de esta forma permitir a un usuario componer consultas más complejas.

7.1 Conclusiones

Hoy en día la información almacenada en las bases de datos juega un papel muy importante en la toma de decisiones realizada por los altos funcionarios de la mayoría de los negocios. Dicha información debe ser precisamente lo que el usuario necesita, pues de lo contrario puede llevar a una toma de decisiones errónea.

En el mercado existen muchos productos orientados a la obtención de información a partir de las bases de datos relacionales. Esto se logra mediante la composición de una consulta en SQL usando una interfaz gráfica. Sin embargo, la mayoría de dichos productos han sido diseñados para su uso por expertos en computación quienes tienen conocimientos sobre el lenguaje de consulta ya mencionado, tal como lo revela el estado del arte presentado en la Subsección 2.2.

Por lo anterior, los usuarios que requieren información de las bases de datos algunas veces quedan excluidos del privilegio de obtener directamente esta información, ya que no cuentan con la debida preparación para realizar consultas a una base de datos por sí mismos; por ejemplo, formulando una consulta en SQL. Considerando esto, en este proyecto de tesis se diseñó e implementó una interfaz intuitiva y fácil de usar para que cualquier usuario que tenga conocimiento sobre el giro de la base de datos pueda componer consultas para obtener la información que desee de una base de datos.

La interfaz propuesta para composición de consultas permite a los usuarios inexpertos componer consultas en SQL sin la necesidad de recibir entrenamiento o tener conocimientos sobre SQL.

Para cumplir con los objetivos descritos en la Subsección 1.3, se diseñó una interfaz con un enfoque completamente diferente de los empleados por otras interfaces (ver Subsección 2.2). La interfaz desarrollada guía al usuario por un proceso de composición de consulta que consta de tres pasos. Esta

interfaz permite al usuario visualizar un fragmento del esquema de BD como una estructura llamada árbol de composición, la cual contiene la información ordenada por niveles de relevancia para que pueda componer su consulta de forma intuitiva.

En conjunción con el árbol de composición, la interfaz proporciona controles con descripciones que explican el contenido de la BD. Además, permite realizar sólo las operaciones más comúnmente usadas por el usuario para componer sus consultas evitando de esta forma que éste cometa errores al momento de componer sus consultas.

Se realizaron pruebas para demostrar la funcionalidad de la interfaz propuesta. Dichas pruebas fueron llevadas a cabo con 17 usuarios que nunca habían utilizado la interfaz a quienes se les proporcionó solamente un manual de operación y 20 consultas. Es importante destacar que **a los usuarios nunca se les proporcionó el esquema de la base de datos ni las consultas correctas en SQL.** Los resultados obtenidos de las pruebas muestran un buen desempeño de la interfaz para la gran mayoría de las consultas planteadas dando un promedio de un minuto para componer una consulta correctamente por medio de la interfaz.

Además, las pruebas realizadas demostraron que los usuarios no experimentaron complicaciones al componer consultas que involucran un máximo de 3 tablas base conectadas directamente, ni tampoco cuando se involucran 2 tablas base conectadas mediante una tabla que implementa una relación muchos a muchos.

Finalmente, es importante mencionar que la técnica usada en esta interfaz de composición de consultas es independiente del lenguaje; por lo tanto, puede ser aplicada a otros lenguajes, tales como inglés, francés, italiano y portugués.

7.2 Trabajos futuros

Considerando que la interfaz presentada en este trabajo permite al usuario componer consultas simples sobre una base de datos compleja, se proponen las siguientes mejoras a la funcionalidad de la misma:

1. Implementar las modificaciones correspondientes para que la interfaz pueda componer las consultas correspondientes a los tipos de consultas del 5 al 9 mencionados en la Subsección 3.3.
2. Implementar el uso de vistas en el árbol de composición.
3. Diseñar e implementar un módulo que permita el uso de vistas y funciones de agregación en la composición de consultas.
4. Implementar el uso de subconsultas mediante la interfaz de composición.
5. Implementar un algoritmo que permita a la interfaz mostrar información relevante para una consulta aun cuando ésta no haya sido solicitada por el usuario; por ejemplo, cuando el resultado de una consulta no muestra al menos una columna que diferencie las filas de la tabla de resultados.
6. Implementar un mecanismo de aprendizaje que permita a la interfaz adaptar la estructura del árbol de composición de acuerdo a las selecciones realizadas por los usuarios al componer consultas mediante la interfaz.
7. Permitir que la interfaz pueda ser usada con diversos motores de bases de datos relacionales (p. ej., MySQL, PostgreSQL, SQLite).

8. Diseñar e implementar un módulo que a partir de una consulta en SQL obtenida mediante el proceso de composición (véase Subsección 4.6), genere una descripción en LN para explicar al usuario el contenido semántico de dicha consulta.

Apéndices

Apéndice A. Descripción de la base de datos ATIS

ATIS (Air Travel Information Service por sus siglas en inglés) es una base de datos relacional (BD) para almacenar información sobre vuelos. Este apéndice presenta los detalles de las tablas de la BD y un diagrama del esquema de BD.

Para describir la base de datos ATIS, primero se presenta el nombre de las tablas, con la descripción usada para este trabajo, y a continuación, por cada columna en la tabla se muestran el nombre de la columna, el tipo de dato de la columna y por último una descripción de la columna.

Tabla: flight y flight_1 **Descripción**: Vuelo de un aeropuerto a otro

Columna	**Tipo**	**Descripción**
flight_code	Numérico	Código de vuelo
flight_days	Texto	Días hábiles
from_airport	Texto	Aeropuerto de origen
to_airport	Texto	Aeropuerto de destino
departure_time	Numérico	Hora de salida
arrrival_time	Numérico	Hora de llegada
airline_code	Texto	Código de aerolínea
flight_number	Numérico	Número de vuelo
class_string	Texto	Código de clase
aircraft_code	Texto	Código de avión
meal_code	Texto	Código de comida
stops	Numérico	Número de escalas
dual_carrier	Texto	Empresa dual
time_elapsed	Numérico	Tiempo de viaje
LLAVE PRIMARIA:	flight_code	

LLAVE FORÁNEA:	flight (aircraft_code) – aircraft (aircraft_code) flight (airline_code) – airline (airline_code) flight (from_airport) – airport (airport_code) flight (to_airport) – airport (airport_code)

Tabla: *airline* y *airline_1* **Descripción:** Aerolínea

Columna	Tipo	Descripción
airline_code	Texto	Código de aerolínea
airline_name	Texto	Nombre de aerolínea
notes	Textoo	Notas

LLAVE PRIMARIA:	airline_code
LLAVE FORÁNEA:	airline (airline_code) – restrict_carrier (airline_code)

Table: *fare* **Description:** Tarifa

Columna	Tipo	Descripción
fare_code	Texto	Código de tarifa
from_airport	Texto	Aeropuerto de origen
to_airport	Texto	Destination airport (Aeropuerto de destino)
fare_class	Texto	Clase de tarifa
fare_airline	Texto	Tarifa de aerolínea
restrict_code	Texto	Código de restricción
one_way_cost	Numérico	Tarifa de viaje sencillo
rnd_trip_cost	Numérico	Tarifa de viaje redondo

LLAVE PRIMARIA:	fare_code
LLAVE FORÁNEA:	fare (restrict_code) – restriction (restrict_code) fare (fare_class) – compound_class (fare_class)

Tabla: *aircraft* **Descripción:** Avión

Columna	Tipo	Descripción
aircraft_code	Texto	Código de avión
aircraft_type	Texto	Tipo de avión
engines	Numérico	Número de motores
category	Texto	Categoría de avión
wide_body	Texto	Fuselaje ancho

wing_span	Numérico	Extensión de alas
length1	Numérico	Tamaño de equipo
weight	Numérico	Peso
capacity	Numérico	Número de asientos
pay_load	Numérico	Capacidad de carga
cruising_speed	Numérico	Velocidad
range_miles	Numérico	Longitud de vuelo
pressurized	Texto	Presurización
LLAVE PRIMARIA:	aircraft_code	

Tabla: *transport* **Descripción:** Servicio de transporte del aeropuerto

Columna	**Tipo**	**Descripción**
transport_code	Texto	Código de transporte
transport_desc	Texto	Descripción de transporte
LLAVE PRIMARIA:	transport_code	

Tabla: *compound_class* **Descripción:** Clase compuesta

Columna	**Tipo**	**Descripción**
fare_class	Texto	Código de clase de tarifa
base_class	Texto	Clase base
class_type	Texto	Tipo de clase
premium	Texto	Es clase premium?
economy	Texto	Es clase económica?
discounted	Texto	Tiene descuento?
night	Texto	Es vuelo nocturno?
season_fare	Texto	Tipo de tarifa de temporada
class_days	Texto	Días en que se aplica la clase
LLAVE PRIMARIA:	fare_class	
LLAVE FORÁNEA: (class_code)	compound_class (base_class) – class_of_service	

Tabla: *ground_service* **Descripción:** Servicio terrestre

Columna	**Tipo**	**Descripción**
city_code	Texto	Código de ciudad

airport_code	Texto	Código de aeropuerto
transport_code	Texto	Código de transporte
ground_fare	Numérico	Tarifa terrestre
LLAVE PRIMARIA:	city_code, airport_code, transport_code	
LLAVE FORÁNEA:	ground_service (transport_code) – transport (transport_code)	

Tabla: *food_service* **Descripción:** Servicio de comida

Columna	**Tipo**	**Descripción**
meal_code	Texto	Código de comida
meal_number	Numérico	Número de comida
meal_class	Texto	Clase de comida
meal_description	Texto	Descripción de comida
LLAVE FORÁNEA:	food_service (meal_code) – flight (meal_code)	

Tabla: *restriction* **Descripción:** Restricción

Columna	**Tipo**	**Descripción**
restrict_code	Texto	Código de restricción
application	Texto	Aplicación
no_discounts	Texto	Descuentos no aplicables
reserve_ticket	Numérico	Boletos en reserva
stopovers	Texto	Escalas
return_min	Numérico	Mínimo de permanencia
return_max	Numérico	Máximo de permanencia
LLAVE PRIMARIA:	restrict_code	

Tabla: *city* **Description:** Ciudad

Columna	**Tipo**	**Descripción**
city_code	Texto	Código de ciudad
city_name	Texto	Nombre de la ciudad
state_code	Texto	Código del estado
time_zone_code	Texto	Código de zona horaria
LLAVE PRIMARIA:	city_code	
LLAVE FORÁNEA:	city (state_code) – state (state_code)	

Tabla: *class_of_service* **Descripción:** Clase de servicio

Columna	Tipo	Descripción
class_code	Texto	Código de clase de servicio
rank	Numérico	Rango
class_description	Texto	Descripción de clase de servicio
LLAVE PRIMARIA:	class_code	

Tabla: *airport_service* **Descripción:** Servicio de aeropuerto

Columna	Typo	Descripción
city_code	Texto	Código de ciudad
airport_code	Texto	Código de aeropuerto
miles_distant	Numérico	Distancia en millas
direction	Texto	Dirección
minutes_distant	Numérico	Distancia en minutos
LLAVE PRIMARIA:	city_code, airport_code	
LLAVE FORÁNEA:	airport_service (airport_code) – airport (airport_code) airport_service (airport_code) – ground_service (airport_code) airport_service (city_code) – ground_service (airport_service) airport_service (city_code) – city (city_code)	

Tabla: *fconnection* **Descripción:** Conexión de vuelo

Column	Type	Description
Connect_code	Numérico	Código de conexión
From_airport	Texto	Aeropuerto de origen
To_airport	Texto	Aeropuerto de destino
Departure_time	Numérico	Hora de salida
Arrival_time	Numérico	Hora de llegada
Flight_days	Texto	Días de vuelo
Stops	Numérico	Escalas
Connections	Numérico	Conexiones
LLAVE PRIMARIA:	connect_code	
LLAVE FORÁNEA:	fconnection (to_airport) – airport (airport_code) fconnection (from_airport) – airport (airport_code)	

Tabla: *connect_leg* **Descripción:** Segmento de conexión

Columna	Tipo	Descripción
Connect_code	Numérico	Código de conexión
Leg_number	Numérico	Número de segmento
Flight_code	Numérico	Código de vuelo
LLAVE FORÁNEA: connect_leg (connect_code) – fconnection (connect_code) connect_leg (flight_code) – flight (flight_code)		

Table: *flight_day* **Descripción:** Días de vuelo

Column	Type	Description
day_mask	Texto	Máscara de días
day_code	Numérico	Código de día
LLAVE PRIMARIA: day_mask, day_code **LLAVE FORÁNEA:** flight_day (day_code) – day_name (day_code)		

Table: *day_name* **Description:** Días

Columna	Tipo	Descripción
day_code	Numérico	Código de día
day_name	Texto	Nombre de día
LLAVE PRIMARIA: day_code		

Table: state **Descripción**: Estado

Columna	Tipo	Descripción
state_code	Texto	Código de estado
state_name	Texto	Nombre de estado
country_name	Texto	Nombre de país
LLAVE PRIMARIA: state_code		

Tabla: *dual_carrier* **Descripción:** Empresas dual

Columna	Tipo	Descripción
main_airline	Texto	Código de aerolínea principal
dual_airline	Texto	Código de empresa dual
low_flight	Numérico	Vuelo económico

high_flight	Numérico	Vuelo costoso
fconnection_name	Texto	Nombre de conexión

LLAVE FORÁNEA: dual_carrier (dual_airline) – airline (airline_code)
dual_carrier (main_airline) – airline (airline_code)

Tabla: *time_zone* **Descripción:** Zona horaria

Columna	Tipo	Descripción
time_zone_code	Texto	Código de zona horaria
time_zone_name	Texto	Nombre de zona horaria

LLAVE FORÁNEA: time_zone (time_zone_code) – city (time_zone_code)

Tabla: *stop1* **Descripción:** Escalas

Columna	Tipo	Descripción
flight_code	Numérico	Código de vuelo
stop_number	Numérico	Número de escala
stop_flight	Numérico	Vuelo con escalas

LLAVE FORÁNEA: stop1 (flight_code) – flight (flight_code)
stop1 (stop_flight) – flight (flight_code)

Tabla: *flight_fare* **Descripción:** Tarifas de los vuelos

Columna	Tipo	Descripción
flight_code	Numérico	Código de vuelo
fare_code	Texto	Código de tarifa

LLAVE FORÁNEA: flight_fare (fare_code) – fare (fare_code)
flight_fare (flight_code) – flight (flight_code)

Tabla: *restrict_carrier* **Descripción:** Restricciones de aerolinea

Columna	Tipo	Descripción
restrict_code	Texto	Código de restricción
airline_code	Texto	Código de aerolínea

LLAVE FORÁNEA: restrict_carrier (restrict_code) – restriction(restrict_code)

Tabla: *restrict_class* **Descripción:** Restricción de clase

Column	Type	Description

restrict_code	Texto	Código de restricción
ex_fare_class	Texto	Tarifa de clase

LLAVE FORÁNEA: restrict_class (restrict_code) – restriction(restrict_code)

El esquema de BD de ATIS se muestra en la Figura A.1. Esta base de datos cuenta con un total de 27 tablas y 123 columnas.

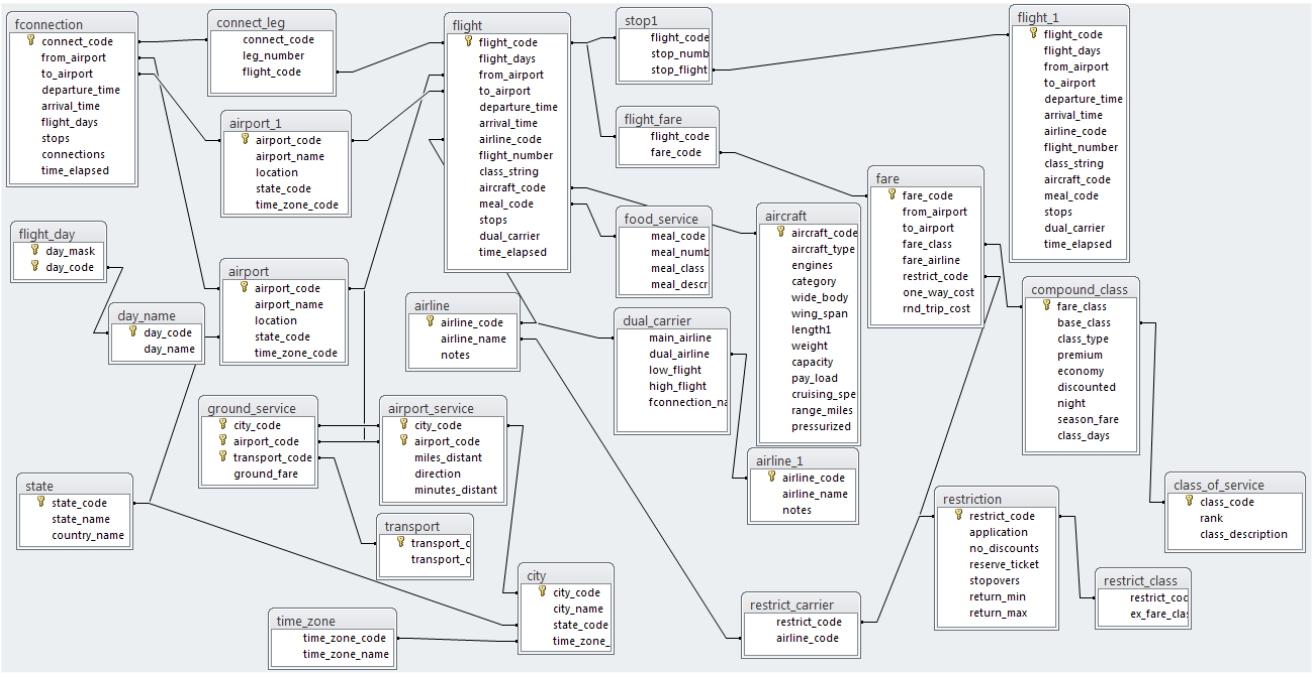


Figura A.1. Esquema de la base de datos ATIS

Apéndice B. Corpus de consultas de la base de datos ATIS

El corpus de consultas presentado en la Tabla B.1 fue diseñado para realizar las pruebas de la interfaz de composición usando la base de datos ATIS descritas en el Capítulo 8.

El corpus de consultas se encuentra dividido en 4 grupos de 5 consultas cada uno, dando un total de 20 consultas ordenadas de menor a mayor de acuerdo a su nivel de dificultad.

Tabla B.1 Corpus de consultas para la base de datos ATIS

No.	Consulta en lenguaje natural
1	¿A qué hora sale el vuelo No. 19?
2	Muéstrame el número de asientos y el peso de un avión tipo BOEING 747SP.
3	¿Cuáles son los códigos del aeropuerto de destino y del aeropuerto de origen del vuelo No. 19?
4	Deseo conocer el tipo de avión y peso de los aviones cuya velocidad sea mayor a 300.
5	¿Cuál es la descripción y clase de comida del servicio de comida cuyo código de comida sea B?
6	¿Cuál es el nombre de la aerolínea del vuelo No. 140?
7	¿Cuál es el nombre del aeropuerto de destino del vuelo número 140?

8	¿Cuáles son los nombres de los aeropuertos que se encuentran en el estado de Colorado?
9	Dame el tipo de avión y el número de asientos del vuelo número 19.
10	Dame los códigos de clase de tarifa y tipos de tarifa de temporada de la clase de servicio con rango 12.
11	Del vuelo No. 16, requiero el número de motores de su avión y el nombre de aerolínea.
12	Dame el nombre de aeropuerto de origen, clase de comida y descripción de comida del vuelo No. 16.
13	Dame el número de vuelo y nombre de aerolínea de los vuelos cuyo nombre de aeropuerto de origen sea GREATER PITTSBURGH.
14	Quiero saber cuál es el nombre de aerolínea del vuelo No. 16 y si su avión es de fuselaje ancho.
15	Dame el número de vuelo, nombre de aerolínea y capacidad de carga del avión de los vuelos cuyo código de aeropuerto de origen sea DFW y código de aeropuerto de destino sea BOS.
16	Dame la tarifa de viaje sencillo del vuelo No. 140.
17	Deseo conocer el máximo y mínimo de permanencia de la aerolínea AMERICAN AIRLINES.
18	¿Cuáles son los nombres de los aeropuertos que sirven a la ciudad de Denver?
19	¿Cuáles son los números de los vuelos que parten desde aeropuertos en el estado de Texas?
20	Dame los números de vuelo y hora de llegada de los vuelos que salen desde San Francisco y llegan antes de las 2100 hrs.

Apéndice C. Descripción del diccionario de información semántica

El diccionario de información semántica (DIS) se usa para almacenar información necesaria para interpretar el contenido de una consulta en lenguaje natural [Aguirre, 2014]. Para este proyecto el DIS se usa para almacenar información sobre el esquema de BD, descripciones de tablas y columnas, así como las definiciones de los tipos de datos.

Por cada tabla en la BD se tiene una descripción que se muestra en el árbol de composición y un tipo de tabla que se usa para construir el árbol de composición.

Tabla: *tipo_tabla_comp* **Descripción**: Tipos de tablas a clasificar

Columna	**Tipo**	**Descripción**
num_tipo	Numérico	Identificador para el tipo de tabla
descripcion	Texto	Descripción del tipo de tabla
LLAVE PRIMARIA:	num_tipo	

Tabla: *tablas_comp* **Descripción**: Descripción de las tablas

Columna	**Tipo**	**Descripción**
nombre_bd	Texto	Nombre de la base de datos
nombre_tabla	Texto	Nombre de la tabla
tipo	Numérico	Número del tipo de tabla
LLAVE PRIMARIA:	nombre_tabla	
LLAVE FORÁNEA:	tipo (tablas_comp) – num_tipo (tipo_tabla_comp)	

Tabla: *columnas_comp* **Descripción**: Descripción de las columnas

Columna	**Tipo**	**Descripción**
nombre_bd	Texto	Nombre de la base de datos
nombre_tabla	Texto	Nombre de la tabla
nombre_columna	Texto	Nombre de la columna
orden	Numérico	Posición en la que aparece la columna en el árbol de composición
tipo	Texto	Identificador del tipo de dato de la columna
formato	Texto	Información de ayuda sobre la columna
descripcion	Texto	Descripción que se muestra en el árbol de composición para la columna
mostrar	Booleano	Indica si la columna se muestra o no.
LLAVE PRIMARIA:	nombre_tabla, nombre_columna	
LLAVE FORÁNEA:	nombre_tabla (columnas_comp) – nombre_tabla (tablas_comp) tipo (columnas_comp) – tipo_dato (datos_comp)	

Tabla: *datos_comp* **Descripción**: Descripción de los tipos de dato

Columna	**Tipo**	**Descripción**
tipo_dato	Texto	Nombre del tipo de dato
menor_txt	Texto	Descripción para el operador de comparación <
mayor_txt	Texto	Descripción para el operador de comparación >
igual_txt	Texto	Descripción para el operador de comparación =

menor_igual_txt	Texto	Descripción para el operador de comparación <=
mayor_igual_txt	Texto	Descripción para el operador de comparación >=
diferente_txt	Texto	Descripción para el operador de comparación <>

LLAVE PRIMARIA: tipo_dato

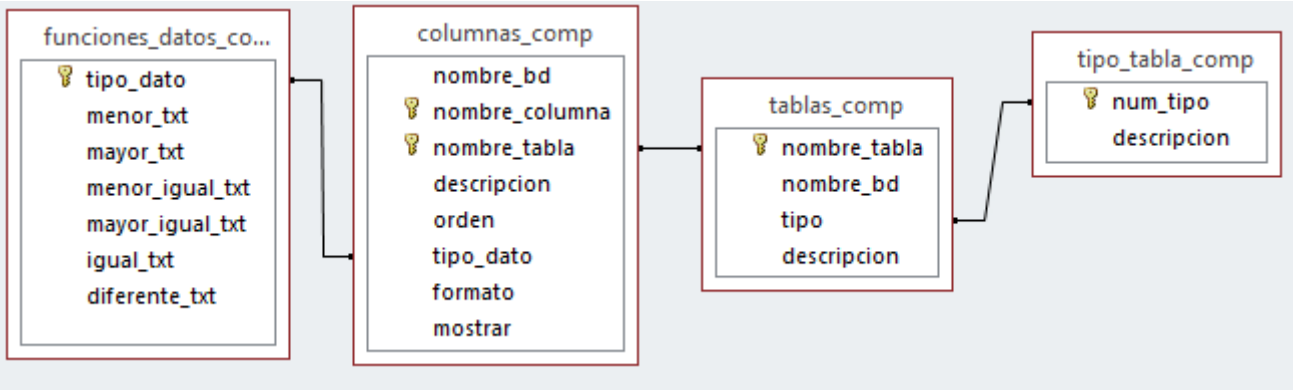


Figura C.1. Esquema del diccionario de información semántica

Referencias

[Aguirre, 2014] M. Aguirre, *Modelo Semánticamente Enriquecido de Bases de Datos para su Explotación por Interfaces de Lenguaje Natural*, tesis de doctorado, División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Ciudad Madero, Cd. Madero, 2014.

[Androutsopoulos, 1995] I. Androtsopoulos, G. Ritchie, P. Thanisch, "Natural Language Interfaces to Database: An Introduction", *Journal of Natural Language Engineering*, Vol. 1, 29-81, 1995.

[Date, 2001] C. Date, *Introducción a los Sistemas de Bases de Datos*, 7a edición, Pearson Educación, 2001.

[González, 2005] J. González, *Traductor de Lenguaje Natural Español a SQL para un Sistema de Consultas a Bases de Datos*, tesis de doctorado, Depto. de Ciencias Computacionales, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Mor., 2005.

[Hallet, 2007] C. Hallet, D. Scott, R. Power, "Composing Questions through Conceptual Authoring", *Computational Linguistics*, Vol. 33, pp. 105-133, 2007.

[Kroenke, 2003] D. Kroenke, *Procesamiento de Bases de Datos*, 8va edición, Pearson Educación, 2003

[Little, 2004] J. Little, M. de Ga, T. Zyer, R. Alhajj, "Query Builder: A natural language interface for structured databases", *ISCIS*, pp. 479-479, 2004.

[May, 2000] A. May, *Herramienta para Consultas Basadas en Ejemplos (QBE) para Multibases de Datos en Internet*, tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Mor., 2000.

[Niesler, 2001] R. Niesler y C. Roux. "Natural Language Understanding in the DACST-AST Dialogue System", *Proceedings of the 12th Annual Symposium of the Pattern Recognition Association of South Africa*, pp. 134-137, 2001.

[Pan, 2010] S. Pan, M. Zhou, K. Houck, P. Kissa, "Natural Language Aided Visual Query Building for Complex Data Access", *Proceedings of the Twenty-Second Innovative Applications of Artificial Intelligence Conference*, pp. 1821-1826, 2010.

[Pazos, 2014] R. Pazos, M. Aguirre, J. Gonzalez, J. Carpio, "Features and Pitfalls that Users Should Seek in Natural Language Interfaces to Databases", *Recent Advances on Hybrid Approaches for Designing Intelligent Systems, Studies in Computational Intelligence*, Vol. 547, pp. 617-630, 2014.

[Rasgado, 1999] F. Rasgado, *Herramienta para Consultas Basadas en Ejemplos (QBE) para una Base de Datos en Internet*, tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Mor., 1999.

[Rojas, 2009] J. Rojas, *Administrador de Diálogo para una Interfaz de Lenguaje Natural a Bases de Datos*, tesis de doctorado, Depto. de Ciencias Computacionales, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Mor., 2009.

[Silberschatz, 2006] A. Silberschatz, H. Korth, S. Sudarshan, *Fundamentos de Bases de Datos*, 5ta edición, McGraw- Hill, 2006.

[Wang, 2007] C. Wang, M. Xiong, Q. Zhou, "PANTO: A Portable Natural Language Interface to Ontologies", *Lecture Notes in Computer Science*, Vol. 4519, pp.473-487, 2007.

[Zhang, 1999] G. Zhang, W. Chu, F. Meng, G. Kong, "Query Formulation from High-level Concepts for Relational Databases", *Proceedings of User Interfaces to Data Intensive Systems*, IEEE Computer Society, pp. 64–74, 1999.