INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

DIVISION DE ESTUDIOS DE POSTGRADO E INVESTIGACIÓN

“Traducción de consultas del lenguaje natural español a SQL que involucran agrupamiento”

PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

ISC ANDRÉS BAUTISTA ALVARADO

DIRECTOR DE TESIS

DR. JOSÉ ANTONIO MARTÍNEZ FLORES

CODIRECTOR DE TESIS

DR. CARLOS ALBERTO OCHOA ORTIZ

CIUDAD MADERO, TAMPS. MÉXICO. ABRIL 2014

Declaración de originalidad

Declaro y prometo que este documento de tesis es producto original de mi trabajo y que no infringe los derechos de terceros, tales como derechos de publicación, derechos de autor, patentes y similares.

Además declaro que en las citas textuales que he incluido y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y las publicaciones.

En caso de infracción de los derechos derivados de este documento de tesis, acepto la responsabilidad de la infracción y relevo de ésta a mi director y codirectores de tesis, así como al Instituto tecnológico de Cd. Madero y sus autoridades.

Abril de 2014, Ciudad Madero, Tamaulipas.

_____

ISC Andrés Bautista Alvarado

DEDICATORIA

La presente tesis es dedicada a:

Jehová, por permitir llegar a esta fase de mi vida y poder completar este proyecto. Por mantenerme siempre de pie y ayudarme en los obstáculos presentes, por darme la mejor de las familias y amigos que siempre me han apoyado.

Mis padres, por ser los mejores y que me han apoyado en todos mis proyectos, mi mamá siempre cuidando de mi bienestar y mi papá siempre dándome el apoyo necesario, los quiero mucho.

Mis hermanas por siempre darme apoyo en lo que necesitaba y por confiar siempre en mí, por sus consejos, sus experiencias y por todo lo que implica el amor de hermano. Las quiero mucho.

AGRADECIMIENTOS

Por todo el apoyo brindado durante este proyecto le doy las gracias a mi Asesor de Tesis el Dr. José Antonio Martínez Flores, quien confió en mí y me apoyó siempre que necesitaba de su ayuda, a mi Co-Director de Tesis el Dr. Carlos Alberto Ochoa Ortiz, quien también deposito su confianza en mí y siempre me apoyó cuando necesité de ayuda.

Agradezco también a mi Comité Tutorial conformado por el Dr. Rodolfo Abraham Pazos Rangel, el Dr. Héctor Joaquín Fraire Huacuja y el MC José Apolinar Ramírez Saldívar, quienes siempre estuvieron pendiente del progreso de mi proyecto y proporcionaban consejos necesarios para que el trabajo de tesis finalizara con éxito.

Muchas gracias a mis compañeros Miguel Ángel y Oscar Manuel por su apoyo y todas las experiencias que pasamos juntos en la Maestría.

Agradecimiento especial para todo el personal académico de la Maestría en ciencias de la computación que siempre nos enseñaron lo mejor y además por todo el apoyo y amistad prestada.

También agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) que me proporcionó el apoyo económico para el desarrollo de esta investigación. De igual manera doy gracias al Instituto Tecnológico de Ciudad Madero por brindarme la oportunidad de superarme profesionalmente con educación de alta calidad.

RESUMEN

En la actualidad el crecimiento de la información ha sido muy rápido, dicha información es almacenada principalmente en bases de datos, por tal motivo, deben desarrollarse herramientas que permitan a los usuarios accederla de manera fácil y sencilla.

En esta tesis se describe el desarrollo de un traductor de consultas de lenguaje Natural español a SQL que involucran funciones de agregación (FAs) y/o agrupamiento. Este trabajo forma parte de un proyecto denominado “Interfaz de Lenguaje Natural Español hacia Bases de Datos para Usuarios de Internet”. Para este proyecto se utilizó como base el modelo semántico propuesto en [1].

Se realizó una clasificación de consultas que involucran FAs y/o agrupamiento utilizando consultas con éstas características de los principales corpus utilizados en la literatura (ATIS, Pubs, Geobase, Northwind y GMISARA).

Los resultados obtenidos muestran un rendimiento del traductor de entre el 83% y 92%.

ABSTRACT

Information has grown faster than a few decades ago, that information is mainly stored in Databases (DBs), and therefore, tools for accessing the information in an easy way should be developed.

In this thesis we described the development from one translator to Natural Language queries in Spanish to SQL that involve Aggregate Functions (AF), and/or grouping. This work is part from a general project called “Spanish Natural Language Interface to Databases for Internet’s users”. In this work we use the semantic model proposed by Aguirre [1].

We realize a classification from queries that involve Aggregate Functions and/or grouping, using queries with these characteristics in main corpus used in literature (ATIS, Pubs, Geobase, Northwind and GMISARA).

Results we obtained show a significant performance for translating queries that involve Aggregate Functions and/or grouping. The translator’s efficiency developed in this work is between 83% and 92%.

LISTA DE FIGURAS

LISTA DE TABLAS

CONTENIDO

CAPÍTULO 3: ESTADO DEL ARTE

CAPÍTULO 4: METODOLOGÍA DE SOLUCIÓN

CAPÍTULO 5: IMPLEMENTACIÓN DEL TRADUCTOR

CAPÍTULO 6: RESULTADOS

CAPITULO 7: CONCLUSIONES Y TRABAJOS FUTUROS

BIBLIOGRAFIA

ANEXOS

Capítulo 1

Introducción

En la actualidad la información sigue en constante crecimiento, los datos en Internet están creciendo rápidamente y se necesita de hardware y software para almacenar dicha información, la cantidad de datos almacenados por empresas se duplica aproximadamente cada 3 años [2]. La información es útil para la toma de decisiones sobre las acciones que se realizan en una empresa o en cualquier situación del mundo real.

La manera normal en la que las personas obtienen información es mediante consultas en lenguaje natural, pero las computadoras no pueden entender este lenguaje. Las Interfaces de Lenguaje Natural para Bases de Datos (ILNBDs) son una alternativa que permite al usuario acceder a información almacenada en bases de datos (BDs), mediante una solicitud en lenguaje natural.

Las ILNBDs se han desarrollado desde los años sesentas y muchos de los problemas que presenta el área de Procesamiento de Lenguaje Natural (PLN) aún no se han podido resolver adecuadamente con técnicas que se han aplicado hasta la fecha [3].

1.1. Antecedentes

Este tema de 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 CENIDET desde septiembre del año 2001 y se ha continuado en el ITCM desde agosto del año 2002. Dicho proyecto tiene como fin implementar una interfaz que permita a usuarios casuales e inexpertos consultar BDs mediante expresiones en lenguaje español no acotado. Una característica importante de esta ILNBD es su independencia de dominio; es decir, su capacidad para interactuar con BDs de diferentes dominios.

La versión de la interfaz desarrollada en ITCM [4] utiliza un método para la traducción de consultas en español a SQL, lo que permite a los usuarios realizar consultas a una base de datos sin necesidad de configuraciones tediosas. El manejo de la independencia del dominio es la característica principal de esta interfaz, y para lograrlo utiliza un preprocesador que genera automáticamente un diccionario de dominio y una técnica de traducción que utiliza sustantivos, preposiciones y conjunciones, ya que estas últimas mantienen el significado en cualquier contexto y no necesitan ser configuradas para un dominio en particular. Las preposiciones y conjunciones son representadas como operaciones usando la teoría de conjuntos. Además, utiliza un grafo etiquetado y una taxonomía de la instrucción SELECT para construir la consulta en SQL. Sin embargo, esta interfaz no permite al usuario un diálogo de aclaración cuando la consulta no puede ser traducida, lo cual limita su capacidad para obtener un mayor porcentaje de aciertos en las consultas traducidas.

La versión de la ILNBD [4] se mejoró con el trabajo de tesis doctoral [3] que implementó procesos de diálogo independientes del dominio para un administrador de diálogo para la ILNBD. Los procesos de diálogo fueron diseñados para funcionar en cualquier base de datos relacional. Para asegurar que los diálogos tuvieran las características anteriormente mencionadas, se formalizó una tipificación de problemas en consultas. Cabe mencionar que solamente se implementaron los procesos de dialogo para algunos de los problemas tipificados.

En la figura 1.1 se muestra el diseño conceptual de la ILNBD desarrollada en [3] con la inclusión de un administrador de diálogo. La interfaz se compone, de forma general, de dos partes: un *procesamiento de la consulta* y un *administrador de diálogo*. El funcionamiento de la interfaz es el siguiente:

I. El usuario selecciona una BD, al hacerlo se lleva a cabo un proceso de configuración de la BD seleccionada (creación de diccionarios: del dominio y de metadatos).

II. El usuario selecciona o escribe una consulta y la ejecuta. Durante el procesamiento se genera información de la consulta que es utilizada para disparar los procesos de diálogo.

III. Con la información generada durante el proceso de traducción, se identifica si la consulta presenta problemas de economía de palabras (módulo *Detección de problema*).

   1. Si existen problemas, se ejecuta el módulo *Selección de procesos de diálogo*, y pasa al punto IV.

   2. Si no existen problemas, continúa el flujo en el punto V.

IV. Se ejecuta el módulo *proceso de diálogo* en base al problema identificado. El *proceso de diálogo* obtendrá información de la BD relacionada con la información faltante, posteriormente crea ventanas para interactuar con el

usuario y solicitar información (flujo A). Con la información proporcionada por el usuario (flujo B), la consulta es modificada y vuelve a procesarse (flujo C). Este proceso se repetirá hasta que la consulta no presente problemas de elipsis semántica.

V. La consulta se traduce a su equivalente en lenguaje SQL.

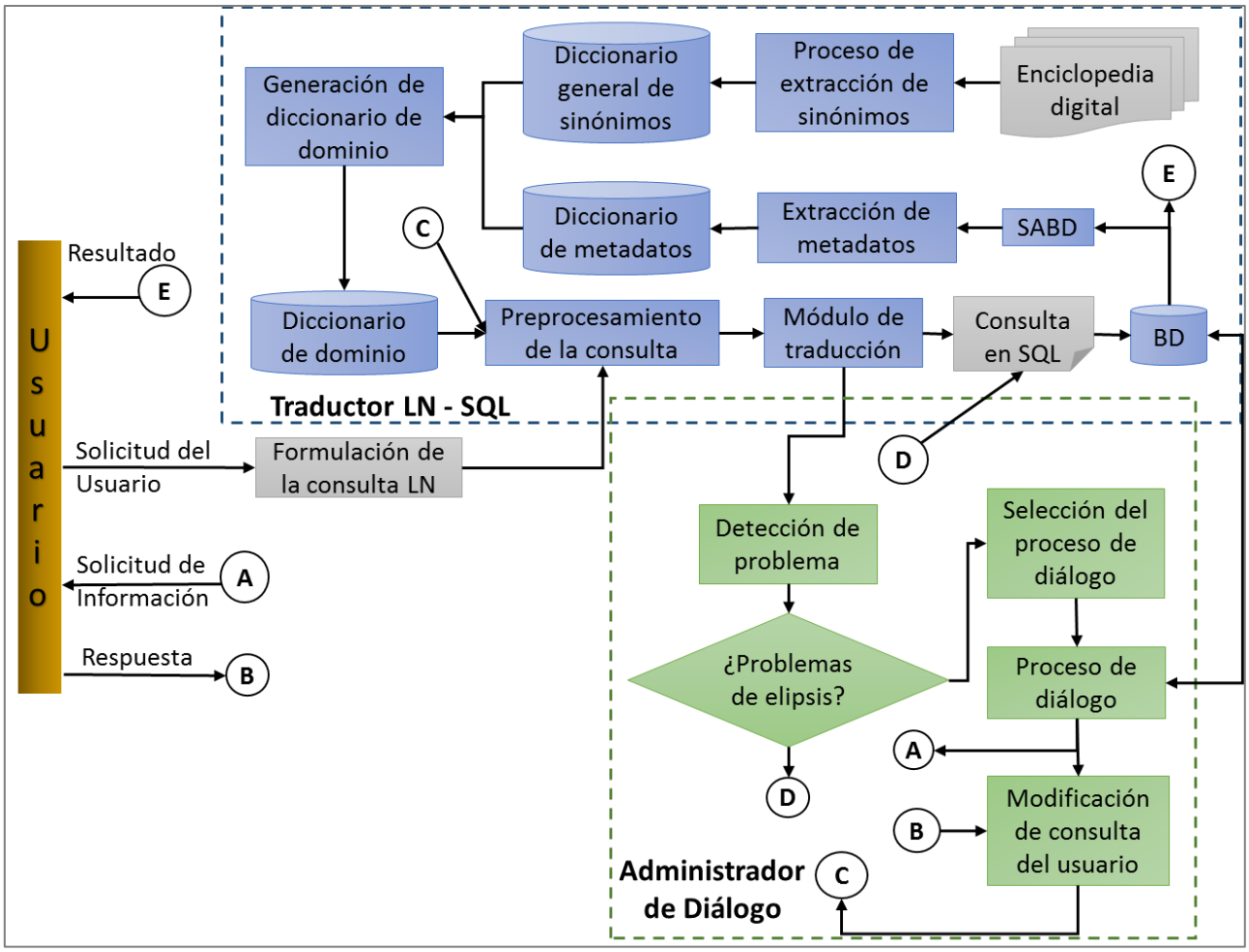


Figura 1.1. Esquema conceptual de una ILNBD con administrador de diálogo

1.2. Descripción del problema

Este proyecto es parte de una ILNBD desarrollada desde el año 2001 por ITCM-CENIDET [5], y de la cual se han generado 2 versiones y una más la está desarrollando el doctorante del ITCM, Marco Antonio Aguirre Lam, la cual se utilizó para desarrollar este proyecto de investigación. Aguirre en su trabajo propone una arquitectura general para ILNBDs [1] (véase Figura 1.2), la cual permite afrontar muchos de los problemas que presentan las ILNBDs.

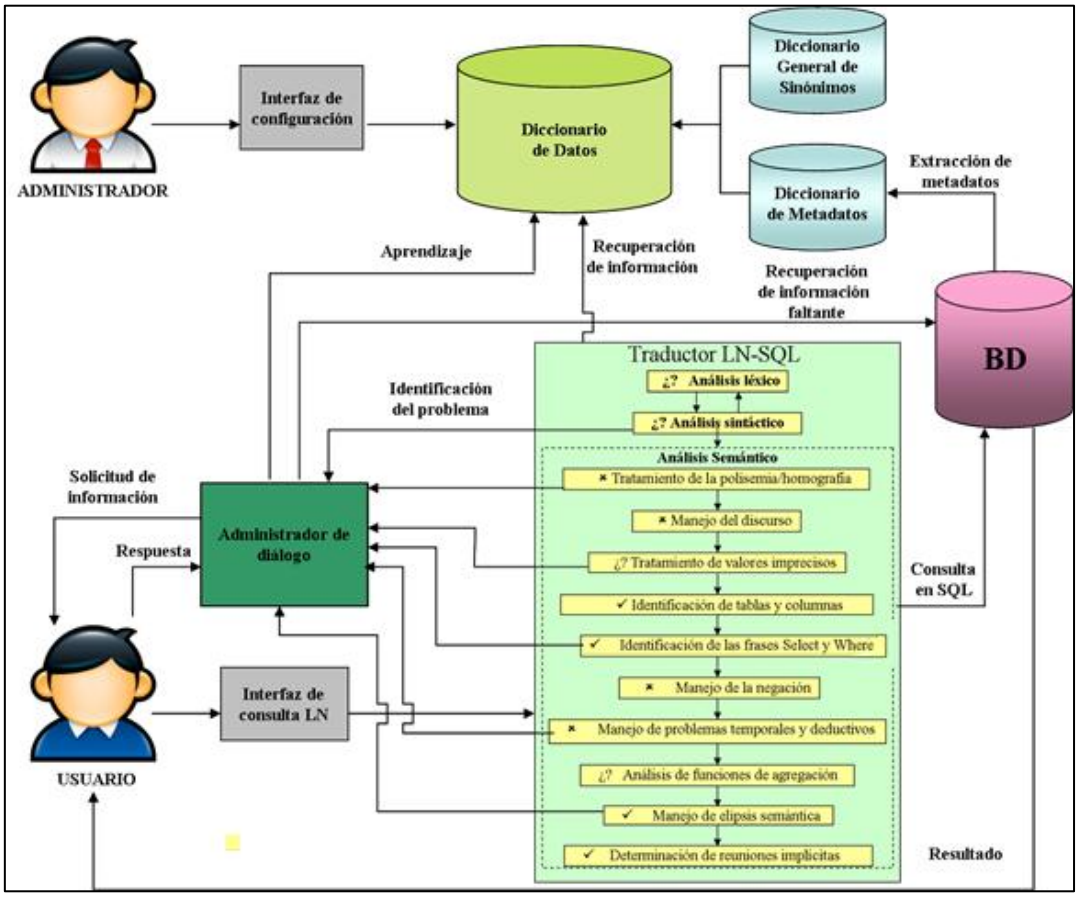


Figura 1.2. Arquitectura general propuesta por Aguirre

Con relación módulo *Traductor de Lenguaje Natural a SQL* de la Figura 1.2, la Figura 1.3 ilustra con un recuadro de línea discontinua los submódulos considerados para el análisis semántico, entre éstos el submódulo *Funciones de agregación y cláusula GROUP BY*, con recuadro en color rojo; elemento clave considerado para implementar el tratamiento de consultas con FA y/o agrupamiento.

La versión de la interfaz de Aguirre no etiqueta las palabras o frases que impliquen el uso de funciones de agregación y/o agrupamiento. Es necesario modificar el analizador léxico de la ILNBD para que el traductor resuelva la problemática anterior.

Otra problemática que surge al procesar consultas que comprendan funciones de agregación y/o agrupamiento, además de identificar consultas con tales características, es que se le deben añadir reglas al analizador sintáctico para que reconozca consultas de este tipo. Así como también se debe actualizar el analizador semántico del traductor para que las traduzca correctamente.

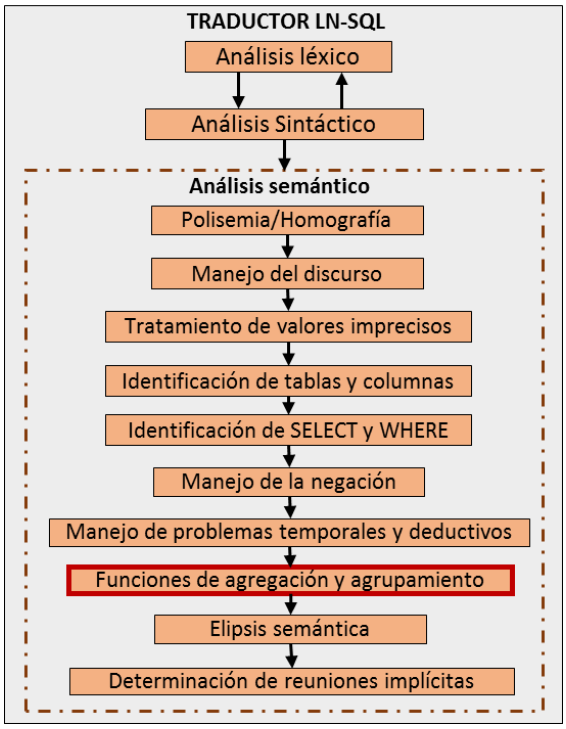


Figura 1.3. Submódulos del traductor LN-SQL.

1.3. Objetivos

1.3.1. Objetivo General

Que la Interfaz desarrollada por el ITCM–CENIDET traduzca consultas que comprendan agrupamientos.

1.3.2. Específicos

- Reconocer consultas en lenguaje natural español que al traducirlas a SQL involucren la cláusula GROUP BY y/o funciones de agregación (FA).

- Analizar la traducción de las consultas anteriores a SQL.

- Diseñar e implementar un módulo para traducir las consultas descritas anteriormente.

1.4. Justificación y Beneficios

Actualmente la mayoría de las empresas utilizan grandes bases de datos para guardar información importante, por ejemplo la de sus clientes, sus proveedores, etc.; desafortunadamente la mayoría de los empleados no tienen conocimientos avanzados de computación, ni saben de comandos SQL para formular las consultas que permitan obtener la información que necesitan, por esta razón es muy importante desarrollar interfaces que proporcionen la información de bases de datos que se requiera por medio de lenguaje natural [6].

A pesar de que las ILNBDs se han desarrollado por varias décadas, aún no se tiene una que conteste correctamente al 100% las consultas que se le hagan; es necesario continuar trabajando en esta área con el propósito de permitir a los usuarios contar con una herramienta completamente fiable para que tomen decisiones importantes de manera segura y confiable [3].

Terminar la ILN que se empezó a desarrollar en el CENIDET permitirá ofrecer a ejecutivos de empresa y a diferentes usuarios una forma flexible de formular consultas y no restringirlos sólo a las ya existentes en los sistemas de consulta que actualmente se tienen desarrollados. Cabe aclarar que en el desarrollo de este proyecto sólo se incrementó la traducción de consultas que involucran agrupamiento y/o funciones de agregación, lo cual permitió mejorar el desempeño de la interfaz.

1.5. Alcances y limitaciones

El alcance que se planteó al inicio de este trabajo de investigación consistió en:

a) Incorporar a la interfaz versión Aguirre un módulo para traducir consultas que involucran funciones de agregación y/o agrupamiento de LN a SQL.

Entre las limitaciones de este trabajo se encuentran las siguientes:

a) El idioma soportado para el traductor de consultas que involucran funciones de agregación y/o agrupamiento es únicamente español.
b) El medio para realizar consultas al traductor es el lenguaje escrito.
c) Las funciones de agregación que el traductor soporta son: MIN, MAX, AVG, SUM y COUNT.
d) La cláusula GROUP BY permite agrupar por más de un atributo.

Además se consideran los siguientes puntos definidos en [3], los cuales son:

e) No proporciona información que no esté explícita, ya que no se planea manejar bases de datos deductivas.

f) Debido a que en SQL una consulta se puede expresar de diferentes maneras, no se considera el problema de transformar una consulta a su equivalente optimizada.

g) Las consultas pueden ser en forma interrogativa e imperativa.

h) Si la consulta contiene un valor compuesto por dos o más palabras, el cual corresponde a una columna de la base de datos, deberá estar escrito entre comillas dobles para poder encontrar su ubicación en la base de datos. Por ejemplo: nombres de personas (“Juan Pérez Fernández”), domicilios, etc.

i) El formato para representar fechas debe ser: dd/mm/aaaa.

Capítulo 2

Marco Teórico

En este capítulo se presenta el soporte teórico de este trabajo de investigación donde se describen los conceptos necesarios involucrados en este trabajo.

2.1. Lenguaje

Conjunto de sonidos articulados o símbolos con que el hombre manifiesta lo que piensa o siente [7]. Cuando se habla de lenguajes se pueden diferenciar dos clases muy bien definidas [8], los lenguajes formales y los lenguajes naturales.

2.1.1. Lenguaje formal

Un lenguaje formal es un lenguaje creado por el hombre, el cual está formado por símbolos y fórmulas y tiene como objetivo fundamental formalizar la programación de computadoras o representar simbólicamente un conocimiento.

2.1.2. Lenguaje natural

Lenguaje hablado o escrito por humanos, opuesto a un lenguaje de programación utilizado para programar o comunicarse con computadoras. Existen dos campos en el estudio del entendimiento del lenguaje natural [9]:

- Entendimiento del lenguaje escrito, que utiliza el conocimiento léxico, sintáctico y semántico del lenguaje, unido a la información o conocimiento del dominio.
- Entendimiento del lenguaje oral, que comprende todo lo del campo anterior junto con toda la fonología.

2.2. Procesamiento de Lenguaje Natural

El Procesamiento de Lenguaje Natural (PLN o NLP en inglés) es un conjunto de técnicas computacionales para analizar y representar naturalmente textos en uno o más niveles de análisis lingüísticos, con el fin de llevar a cabo el procesamiento del lenguaje como un humano para un rango de tareas y aplicaciones [10].

2.3. Base de datos

Una base de datos (BD) 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 [11].

- Una base de datos representa algún aspecto del mundo real, en ocasiones llamado minimundo o universo del discurso; las modificaciones de éste se reflejan en la BD.
- 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 prueba con datos para un propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a dichos usuarios.

Las bases de datos computarizadas se pueden crear y mantener con un grupo de programas de aplicación escritos específicamente para esa tarea, o bien mediante un sistema administrador de bases de datos.

Un sistema administrador de bases de datos es un conjunto de programas que permite a los usuarios crear y mantener una base de datos. Por tanto, un sistema administrador de bases de datos es un software de propósito general que facilita el proceso de definir, construir y manipular bases de datos para diversas aplicaciones.

2.4. Interfaz de Lenguaje Natural

Las interfaces de lenguaje natural son mecanismos de comunicación entre una persona y una máquina a través de lenguaje natural. Por lo general, esta comunicación es bidireccional, es decir, de tipo pregunta-respuesta [6]. El diagrama general de una ILN se presenta en la Figura 2.1.

2.5. Interfaz de Lenguaje Natural para Bases de Datos

Una ILNBD es un sistema que permite al usuario acceder a la información almacenada en una BD formulando una solicitud en lenguaje natural [12].

La ILNBD es un sistema intermedio entre el usuario y la información contenida en una BD, la interfaz recibe una solicitud en lenguaje natural realizada por el usuario, esta consulta es procesada por la interfaz y genera una instrucción en SQL. Posteriormente el comando SQL es consultado sobre la BD y este resultado se retorna al usuario.

2.6. Structured Query Language

SQL (Structured Query Language, en español Lenguaje de Consultas Estructurado) 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, modificarlos y especificar restricciones de seguridad [13].

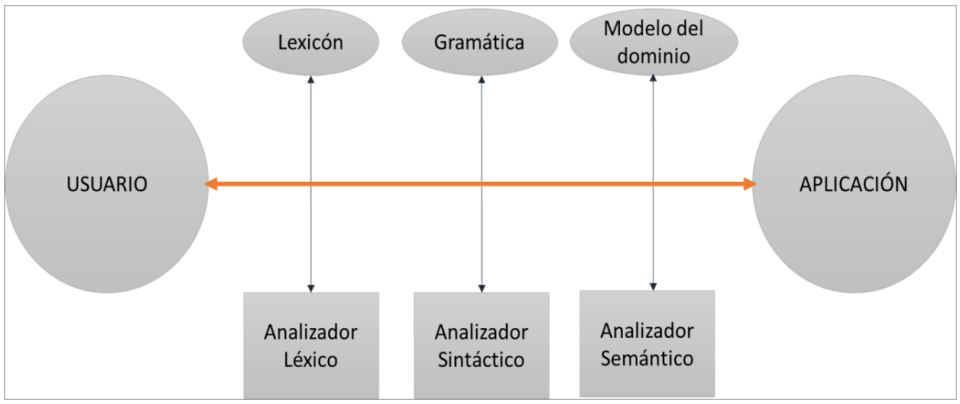


Figura 2.1. Arquitectura general de una ILN

2.7. Funciones de agregación en SQL

Las funciones de agregación son funciones que toman una colección de valores como entrada y producen un único valor de salida. SQL proporciona cinco funciones de agregación primitivas (véase Figura 2.1) que permiten realizar cálculos con la información de las tablas de las BDs.

Tabla 2.1. Funciones de agregación en SQL

Función	**Descripción**
COUNT	Retorna el número total de filas seleccionadas.
SUM	Retorna la suma de valores de una columna.
MIN	Retorna el valor mínimo de una columna.
MAX	Retorna el valor máximo de una columna.
AVG	Retorna el valor promedio de una columna.

2.7.1. Función de agregación COUNT

La función COUNT permite contar el número de filas de una tabla determinada.

La sintaxis básica es:

```
SELECT COUNT ("nombre_columna")
FROM      "Nombre_Tabla"
```

2.7.2. Funciones de agregación MAX y MIN

La función MAX devuelve el valor mayor de la columna que se pasa como parámetro. La función MIN devuelve el valor menor de la columna.

La sintaxis básica es:

```
SELECT [MAX | MIN] ("nombre_columna")
FROM      "Nombre_Tabla"
```

2.7.3. Funciones de agregación AVG y SUM

La función SUM retorna la suma de los datos de la columna que se pasa de parámetro, la función AVG calcula y retorna el valor promedio de los datos que se encuentran en la columna que se pasa de parámetro. Las funciones AVG y SUM realizan cálculos con tipos de dato numéricos, si la columna que se pasa como parámetro tiene un tipo de dato diferente, el SMBD despliega un mensaje de error al ejecutar la instrucción.

La sintaxis básica es:

```
SELECT [AVG | SUM] ("nombre_columna")
FROM      "Nombre_Tabla"
```

2.8. Cláusula de agrupación GROUP BY en SQL

Las funciones de agregación mencionadas anteriormente también se pueden utilizar con la cláusula de agrupación GROUP BY para generar consultas más detalladas.

La cláusula de agrupación GROUP BY se utiliza junto con las funciones de agregación mencionadas anteriormente para generar consultas más detalladas y específicamente consultas donde se desea crear un agrupamiento de los datos contenidos en una tabla.

Para utilizar la cláusula de agrupación GROUP BY se debe al menos incluir alguna función de agregación; entonces, se necesita hacer dos cosas primordiales: 1) aplicar alguna función de agregación a los atributos que se desean recuperar y 2) utilizar la cláusula GROUP BY para agrupar por registros específicos.

La sintaxis básica SQL correspondiente es:

```
SELECT      "Nombre1_columna", SUM ("nombre2_columna")
FROM        "Nombre_Tabla"
GROUP BY    "nombre1-columna"
```

Consideremos la tabla *informacion_de_tienda* (Véase tabla 2.2)

Tabla 2.2. Datos de tabla *informacion_de_tienda*

Nombre_tienda	Venta ($)	Fecha
Los Ángeles	1500	05-Jan-1999
San Diego	250	07-Jan-1999
Los Ángeles	300	08-Jan-1999
Boston	700	08-Jan-1999

Para conocer el número de tiendas de la tabla 2.3 se ejecuta la siguiente consulta en SQL.

```
SELECT COUNT (nombre_tienda)
FROM   Informacion_de_tienda
```

El resultado de ejecutar la consulta anterior es 4, el total de registros de la tabla.

Si se requiere el total de ventas de cada tienda es necesario utilizar la cláusula GROUP BY para obtener la información solicitada, la instrucción para obtener esta información es la siguiente:

```
SELECT    nombre_tienda, SUM (venta)
FROM      informacion_de_tienda
GROUP BY  nombre_tienda
```

El resultado de ejecutar la instrucción anterior se muestra en la tabla 2.3.

Tabla 2.3. Resultado de ejecutar una consulta con agrupamiento

Nombre_tienda	SUM (venta)
Los Ángeles	1800
San Diego	250
Boston	700

Capítulo 3

Estado del Arte

En este capítulo se muestran los trabajos que se encontraron en la literatura y se relacionan con este proyecto de tesis.

3.1. MASQUE/SQL

MASQUE es una ILN desarrollada por la Universidad de Edinburgh para recuperar información de BDs. MASQUE traduce consultas en LN inglés a consultas en *Prolog*. En [12] se desarrolló una modificación a MASQUE para permitir realizar consultas a BDs relacionales, la versión modificada de MASQUE recibió el nombre de MASQUE/SQL.

La modificación que se le realizó a MASQUE permitió responder consultas que involucran funciones de agregación así como de agrupamiento. Un ejemplo de las traducciones que realiza MASQUE después de que se modificó, se muestra en la figura 3.1.

```
QAS> what is the average area of the countries in each continent?
LQL Query :
answer([B,C]) :-  continent(B) & F = setof D:[E]  area(D,E)
& country(E) & in(E,B) & av(C,F)

SQL query :
      SELECT DISTINCT rel1.arg1, avg(rel2.arg1)
      FROM continent#1 rel1, area#2 rel2, country#1 rel3, in#2 rel4
      WHERE rel3.arg1 = rel2.arg2 AND rel4.arg1 = rel2.arg2 AND
      rel4.arg2 = rel1.arg1 GROUP BY rel1.arg1

africa 234090
america 496712
asia 485621
australasia 543940
europe 58808.9
5 solution(s)
Total time used: 5.35 sec.
```

Figura 3.1: Ejemplo de traducción de la interfaz MASQUE.

3.2. VICENT

En [14] muestran el desarrollo de una interfaz de consultas para usuarios chinos; debido a que las interfaces que se desarrollan son principalmente para usuarios de habla inglesa crearon VICENT, ya que para un usuario chino es muy difícil aprender comandos en lenguaje inglés.

La traducción de la interfaz se centra en realizar operaciones de unidad, esto es, que cada consulta formulada en la interfaz se puede descomponer en varias subconsultas sin afectar el significado de la consulta inicial. Para el proceso de traducción se definen diferentes tipos de reglas sobre consultas básicas en lenguaje chino.

Las reglas están clasificadas en 6 categorías { proyección, restricción, relación, división, intersección y unión }, además se da soporte a las funciones de agregación y a la cláusula GROUP BY.

En la sección 4.2 de [14] se muestra una traducción en la que se utiliza la cláusula GROUP BY.

3.3. IDICULA

Diseño y desarrollo de un sistema basado en marco adaptable para procesamiento del idioma dravidian. Para el idioma dravidian existen cerca de 17 lenguas familiares que son habladas por la gente del sur de la India. Estas lenguas son lenguajes independientes del orden de palabras. Dado que la mayoría de las gramáticas computacionales existentes son gramáticas posicionales, éstas no son adecuadas para el análisis y la comprensión de lenguajes dinámicos del orden de palabras. En la tesis doctoral [15] se muestra el desarrollo de un sistema basado en marco adaptable para el tratamiento de lenguas dravidian. Para el desarrollo del sistema utilizaron como base la lengua malayalam (típica de la familia dravidian y del idioma nativo de Kerala).

Para la traducción de consultas del idioma dravidian a sentencias SQL, primero realizan una traducción del idioma dravidian al inglés, posteriormente del idioma Inglés se traduce la consulta a SQL. El método que utilizan para la traducción se basa en un conjunto de patrones y reglas que forman el esqueleto de las sentencias que se pueden generar y traducir.

El sistema de traducción se incorporó a una interfaz de lenguaje natural donde los usuarios pueden realizar fácilmente sus consultas. La Figura 3.2 muestra el diagrama esquemático del sistema de ILN Idicula.

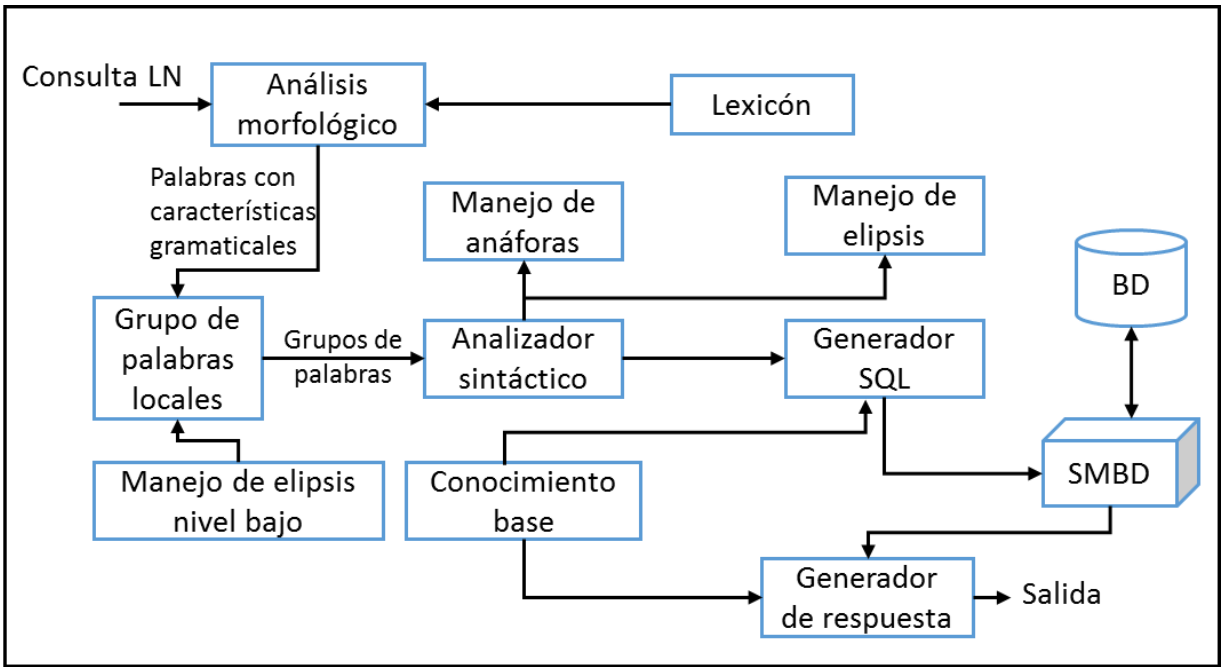


Figura 3.2. Diagrama esquemático de la ILN Idicula

En la Figura 3.3 tomada de la tesis [15] se muestra una consulta y su respectiva traducción del idioma Dravidian a SQL, mostrando la traducción intermedia del idioma inglés.

14.100ൽ കൂടുതൽ വിദ്യാർത്ഥികൾ ഉള്ള ഡിപ്പാർട്ടുമെന്റ് ഏതെല്ലാം?

100Il kutuw2l vixyA2RwWikL uLL dipARRum2neRz EwellAM

(Which departments have more than 100 students ?)

```
SELECT department.deptname ,COUNT(student.studname)
FROM department, student
WHERE student.deptid = department.deptid
GROUP BY department.deptname
HAVING COUNT(student.studname) > 10
```

Department.deptname

1. mawwmaRRikzs

Figura 3.3: Ejemplo de traducción del sistema de ILN Idicula [15].

3.4. OWDA

Interfaz basada en conversación de Lenguaje Natural para Bases de datos Relacionales. El trabajo expuesto en [16] presenta un nuevo enfoque para la creación de interfaces basadas en una conversación en lenguaje natural a bases de datos relacionales combinando agentes conversacionales y árboles del conocimiento. Se trata de un *framework* que facilita el desarrollo de ILN basadas en conversaciones.

El *framework* en [16], presenta un ejemplo de una sesión conversacional entre la ILN y el usuario, cabe destacar que este ejemplo retorna como resultado una sentencia SQL que incluye la cláusula GROUP BY.

3.5. DaNaLIX

El desarrollo de una ILN de dominio adaptativo para consultas en XML se describe en [17], en donde se realizó una adaptación de NaLIX (véase figura 3.4), una interfaz genérica para consultas XML en lenguaje natural. DaNaLIX aprovecha el conocimiento de 3 dominios para mejorar la traducción de consultas y permite la portabilidad.

Con la incorporación de los conocimientos de dominio, DaNaLIX ayuda a reducir la necesidad de reformular consultas que contienen términos con la semántica de dominio. Por ejemplo, la consulta *"What is the most expensive book in year 2000?"* no la respondía correctamente NaLIX, sin embargo DaNaLIX toma ventaja del conocimiento existente y la traduce correctamente.

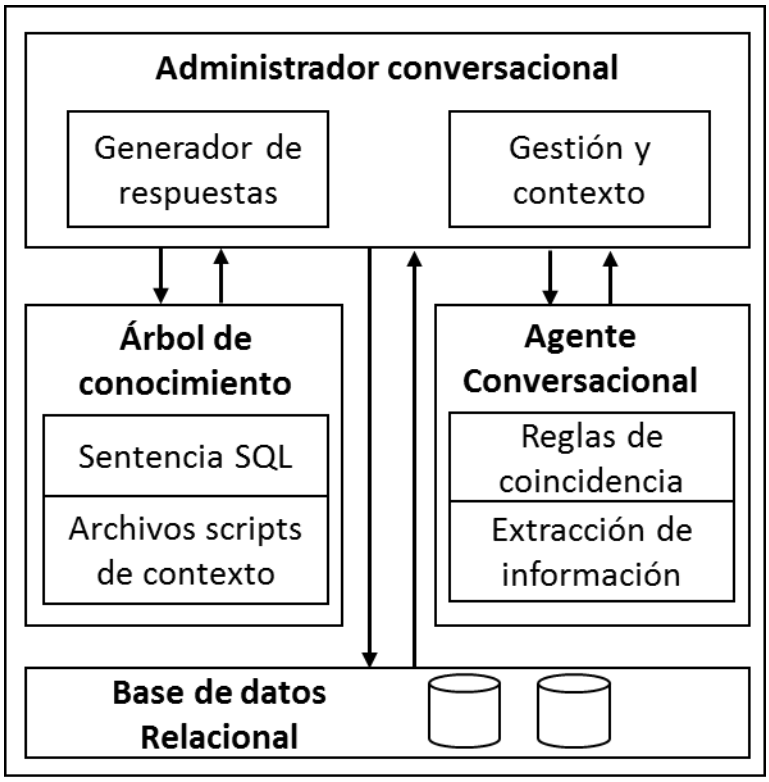


Figura 3.4. Arquitectura de NaLIX.

La arquitectura de DaNaLIX se puede observar con detalle en la figura 3.5.

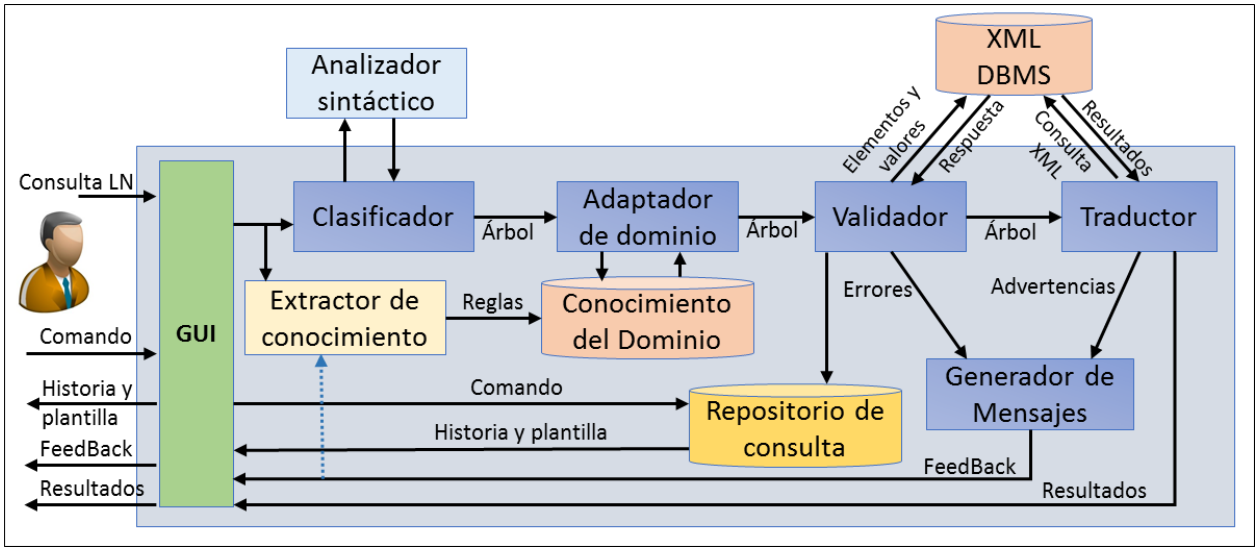


Figura 3.5. Arquitectura de DaNaLIX

3.6. ELF

ELF es un sistema comercial de procesamiento de lenguaje natural para bases de datos. Este sistema es desarrollado por ELF Software Co. ELF es una interfaz que puede trabajar con Microsoft Access y Visual Basic [18].

ELF realiza el procesamiento a partir de un diccionario de conocimiento de la Base de Datos a la cual se requiere consultar.

3.7. SNL2SQL

Traducción de expresiones estadísticas en español a SQL, en [19] se describe un sistema para el reconocimiento de expresiones estadísticas en español (total, subtotal, suma, promedio, máximo, mínimo, etc.) en consultas en lenguaje natural realizadas a una base de datos, posteriormente se traducen a SQL. Se trata de un módulo de una ILNBD llamada SNL2SQL, en fase de desarrollo. El sistema maneja tres tipos de preguntas en español relacionadas con las expresiones estadísticas como se muestra en la tabla 3.1.

Tabla 3.1. Tipos de consultas y sentencias asociadas

Tipo	**Consulta**	**Sentencia SQL**
1	¿Cuál es el total de salarios?	Select sum(Empl.salary) From Empl
2	¿Cuántos empleados se tienen por cada departamento?	Select Empl.Workdept, count(*) From Empl Group by 1 Order by 2 Desc
3	¿Quién gana más salario de los empleados?	Select Empl.Empno, Sum(Empl.salary) From Empl Group by 1 Order by 2 Desc

Asocia a cada función de agregación un conjunto de expresiones estadísticas y un tipo de dato como se muestra a continuación:

SUM ([DISTINCT] X) = {Total, suma, sumatoria, subtotal,…} [Numérico]
AVG ([DISTINCT] X) = {Promedio, media,…} [Numérico]
MIN ([DISTINCT] X) = {Menos, menor, mínimo,…} [Cualquier Tipo]
MAX ([DISTINCT] X) = {Más, mayor, máximo,…} [Cualquier Tipo]
COUNT ([DISTINCT] X) = {Cuantos, cantidad,…} [Valores]
COUNT (*) = {Más, mayor, máximo,…} [Tuplas]

Tomando en cuenta las características de las consultas y habiendo implementado su algoritmo se evaluó el sistema de traducción. Los resultados para los tipos de preguntas mostrados en la tabla 3.1 son:

- Consultas del tipo 1: **81.48 %.**
- Consultas del tipo 2: **76.47 %.**
- Consultas del tipo 3: **76.46 %.**

3.8. Resumen del Estado del Arte

A lo largo de los años se han desarrollado diversas ILNBDs y sin embargo en la actualidad no existe alguna para el idioma español que responda correctamente a consultas que involucren agrupamiento y que funcione para diferentes dominios.

En la Tabla 3.2 se muestran las principales interfaces desarrolladas para el procesamiento de consultas a BDs mediante lenguaje natural.

Tabla 3.2. Trabajos e ILNBD para traducir consultas de LN a SQL.

INTERFAZ	FAs	Cláusula Group By	Independiente de dominio	Idioma	Año
MASQUE	✓	✓	✓	Inglés	1992
VINCENT	✓	✓	✓	Chino	1995
TAMIC	X	X	X	Inglés	1996
IDICULA	✓	✓	✓	Dravidian	1999
PRECISE	X	X	✓	Inglés	2003
InBase	X	X	✓	Inglés	2003
NLPQC	X	X	✓	Inglés	2005
CENIDET	X	X	✓	Español	2005
WYSIWYM	X	X	✓	Inglés	2006
OWDA	✓	✓	✓	Inglés	2007
C-PHRASE	X	X	✓	Inglés	2008
Rojas	X	X	✓	Español	2009
STK	X	X	✓	Inglés	2010
ELF	✓	✓	✓	Inglés	2010
SNL2SQL	**‡**	**‡**	**X**	**Español**	**2010**
Este proyecto	✓	✓	✓	**Español**	**2012**

‡ *= No se incluyen todas / Uso limitado*

Capítulo 4

Metodología de Solución

En este capítulo se presenta la metodología utilizada para dar solución al problema descrito en este trabajo.

4.1. Metodología de solución

La metodología utilizada para lograr los objetivos de este trabajo se basa en las siguientes actividades.

1. Analizar la interfaz versión Aguirre [6].
2. Analizar consultas que involucran FA y/o agrupamiento.
3. Realizar una clasificación de consultas que involucran FA y/o agrupamiento.
4. Diseño e implementación del traductor de consultas con FA y/o agrupamiento.

4.2. Análisis de la interfaz versión Aguirre

Para realizar un buen diseño de nuestra propuesta de solución a la problemática planteada fue necesario conocer la interfaz que se utilizó como base para el desarrollo de este proyecto. Se realizó un análisis del funcionamiento de la ILNBD desarrollada por ITCM-CENIDET [3].

4.3. Análisis de consultas que involucran FA(s) y/o agrupamiento

Esta actividad se realizó para determinar los diferentes tipos de consultas que involucran agrupamiento y/o FA(s) existentes en los corpus de las bases de datos que se analizaron.

Algunos ejemplos de consultas del corpus de la BD GEOBASE se muestran en la tabla 4.1, la tercera columna muestra la FA y/o agrupamiento que requiera la traducción a SQL.

Tabla 4.1. Ejemplos de consultas del corpus GEOBASE

	Consulta	**Requiere uso de**
1	Which rivers run in California state? ¿Qué ríos corren en el estado de California?	Ninguna
2	What is the biggest state by area? ¿Cuál es el estado más grande por área?	MAX
3	How many cities are by state? ¿Cuántas ciudades hay por estado?	COUNT y GROUP BY
4	How many mountains are in Ohio state? ¿Cuántas montañas hay en el estado de Ohio?	COUNT

En la tabla 4.2 se muestran los 5 corpus analizados en este proyecto de tesis, el número de consultas que los conforman, así como también el número de consultas que al traducirlas a SQL requieren una FA y/o agrupamiento.

Tabla 4.2. Corpus de consultas con FA y/o agrupamiento

	Geobase	**Northwind**	**Pubs**	**GMISARA**	**MCC**
No. Consultas	250	198	70	3	134
MAX	46	0	0	0	19
MIN	18	0	1	0	22
COUNT	14	1	5	0	24
AVG	0	0	0	1	6
SUM	4	0	1	0	4
Agrupamiento	0	0	0	2	16

4.3.1. Características de Consultas con FA(s)

Las consultas con funciones de agregación regresan como resultado un único valor para cada función que la consulta presente, el valor es obtenido según la función de agregación (MAX, MIN, AVG, SUM, COUNT).

Otra información importante obtenida durante el análisis de consultas que involucran funciones de agregación son los patrones asociados a éstas, los cuales se describen en la tabla 4.3.

Tabla 4.3. Patrones de consultas que involucran funciones de agregación

#	**Descripción**
1	La columna a la que hace referencia la FA se encuentra antes de la [palabra/frase clave]. Ej. Dame el **área** **combinada** de todos los estados.
2	La columna a la que hace referencia la FA se encuentra después de la [palabra/frase clave] Ej. ¿Cuál es el río con **mayor** **longitud**?
3	Las 2 columnas a las que hace referencia la FA se presentan entre una conjunción y pueden localizarse antes o después de la [palabra/frase clave] Ej. Dame el **área** y la **población** **total** de todos los estados. * Ej. Dame el **total** del **área** y de la **población** de los estados. *
4	La columna a la que hacen referencia 2 FAs separadas por una conjunción se presenta antes o después de las [palabras/frases claves] Ej. ¿Cuál es la **altura** **menor** y **mayor** de todas las montañas? * Ej. Dime cuál es la **menor** y la **mayor** **edad** de los alumnos. *
5	La(s) columna(s) a la(s) que hacen referencia varias FAs se presentan según las descripciones anteriores. Ej. Dame la **mayor** **área** y la **población** **máxima** de los estados. * Ej. Dame el **área** **total** de todos los estados y la **altura** **promedio** de sus montañas. *

* *Consulta inventada para ejemplificar*

4.3.2. Características de consultas con agrupamiento

Las consultas con agrupamiento permiten obtener información agrupada por columnas específicas, este tipo de consulta debe realizarse con la cláusula GROUP BY y debe al menos contener alguna de las funciones de agregación del lenguaje SQL.

En el lenguaje natural las consultas que involucran la cláusula GROUP BY se pueden encontrar de acuerdo a patrones que se obtuvieron durante el análisis de las consultas de este tipo. Los patrones asociados a las consultas que involucran agrupamiento se presentan en la tabla 4.4.

Tabla 4.4. Patrones de consultas que involucran agrupamiento

#	**Descripción**
1	La columna a la que hace referencia la cláusula de agrupamiento (GROUP BY) se encuentra después de la [palabra/frase clave]. Ej. ¿Cuántas ciudades hay **por estado**?
2	Las 2 columnas a las que hace referencia la cláusula GROUP BY se presentan entre una conjunción y se localizan después de la [palabra/frase clave] Ej. Dime la cantidad de alumnos **por proyecto** y **carrera**. *

* *Consulta inventada para ejemplificar*

4.4. Clasificación de consultas que involucran FA y/o agrupamiento

Realizar un análisis de las consultas que involucran funciones de agregación y/o agrupamiento permitió encontrar características de este tipo de consultas, dichas características se tomaron en cuenta para crear una clasificación de consultas en LN que al traducirlas a SQL involucran alguna FA y/o agrupamiento.

La tabla 4.5 muestra la propuesta para clasificar consultas que involucran alguna FA y/o agrupamiento, la cual se realizó después de concluir el análisis de consultas de los corpus analizados. Para las categorías 4 y 5 se inventaron las consultas para ejemplificar.

Tabla 4.5. Clasificación de consultas que involucran FA y/o agrupamiento

Cat.	Descripción
1	Información explícita de función(es) de agregación FA(s) en la cláusula SELECT Consultas que presentan alguna FA y el nombre de la columna a la que hace referencia.
	¿Cuál es el **área combinada** de todos los estados? ¿Cuál es la **población combinada** de todos los estados?
2	Información implícita de FA(s) en la cláusula SELECT Consultas que presentan FA(s) pero no especifican la columna a la que hace referencia.
	¿Cuál es la ciudad **más grande** en el estado de Arizona? ¿Cuál es el río **más largo**?
3	Información explícita de columna(s) en la cláusula GROUP BY Consultas que presentan cláusula GROUP BY y el nombre de la columna a la que hace referencia el agrupamiento.
	¿Cuántas ciudades hay **por cada estado**? Total de estudiantes **por sexo**
4	Información implícita de columna(s) en la cláusula GROUP BY Consultas que presentan la cláusula GROUP BY pero no especifican la columna a la que hace referencia el agrupamiento.
	Muestra el número de alumnos por su **estudio** Lista la edad promedio de los alumnos por **carrera**
5	Agrupamiento con condición (Cláusula HAVING) Consultas que involucran agrupamiento y además contienen condición en alguna de las funciones de agregación.
	Total de estudiantes por carrera donde **el promedio de edad** sea mayor a **20 años** Lista las carreras por nombre donde **el total de alumnos** sea mayor de **300**
6	FA(s) en subconsultas (Cláusula WHERE) Consultas que requieren el uso de FA(s) como condición en la cláusula WHERE para mostrar el resultado correspondiente.
	¿Cuál es el *estado* con la **mayor área**? ¿Cuál es el la *montaña* con la **mayor altura**?

4.5. Diseño del traductor de consultas con FA y/o agrupamiento

Después de analizar la traducción de lenguaje natural a SQL de consultas que involucran FA(s) y/o agrupamiento(s), se realizó el diseño.

La arquitectura general de la ILNBD versión de Aguirre [1] una vez que finalizó este trabajo de tesis se muestra en la Figura 4.1; los colores de los recuadros representan lo siguiente:

- azul (módulo añadido)
- rojo (módulo modificado)
- verde (módulo sin afectación)

La arquitectura presentada en la Figura 4.1, se divide en cuatro capas; Análisis Léxico, Análisis Sintáctico, Análisis Semántico y Traducción SQL. Cada capa esta puede estar conformada por módulos y éstos a su vez pueden estar formados por submódulos.

Durante el análisis léxico se etiquetan la(s) palabra(s) y/o frase(s) de la consulta en LN que es introducida a la interfaz que indique el uso de FA(s) y/o agrupamiento(s); de acuerdo a las palabras y frases que se obtuvieron en el análisis realizado en las actividades anteriores. La tabla 4.6 muestra las palabras y frases que se obtuvieron en dicho análisis.

Tabla 4.6. Palabras y frases para identificar elementos en una consulta

Palabra o frase	Función	Palabra o frase	Función	Palabra o frase	Cláusula
Cantidad media	Avg	El máximo	Max	Por cada	Group by
Media	Avg	El mayor	Max	Por	Group by
Medio	Avg	La máxima	Max	Agrupado por	Group by
Promedio	Avg	Más alto	Max	Agrupada por	Group by
Valor medio	Avg	Más amplio	Max	Clasificada por	Group by

Valor promedio	Avg	Más grande	Max	Para cada	Group by
Cantidad de	Count	Máxima	Max	Clasificado por	Group by
Cuántas	Count	Máximo	Max	Clasificado en	Group by
Cuántos	Count	Con más	Max	En cada	Group by
Cuenta de	Count	Mayor	Max	De cada	Group by
La cantidad	Count	Superior	Max	Agrupado en	Group by
La cuenta	Count	El total	Sum	Agrupada en	Group by
Número de	Count	La sumatoria	Sum	Suma total	Sum
Inferior	Min	Subtotal	Sum	Sumatoria	Sum
Más bajo	Min	Subtotal de	Sum	Total	Sum
Más chico	Min	Suma (de)	Sum	Mínima(o)	Min

Dentro del análisis semántico se actualizó el módulo ***Identificación de las frases SELECT y WHERE*** para permitir analizar la semántica de consultas que involucran funciones de agregación y/o agrupamiento (cláusula GROUP BY). Además se actualizó el módulo correspondiente a la capa **Traducción SQL**, el cual convierte la consulta en LN a una instrucción SQL.

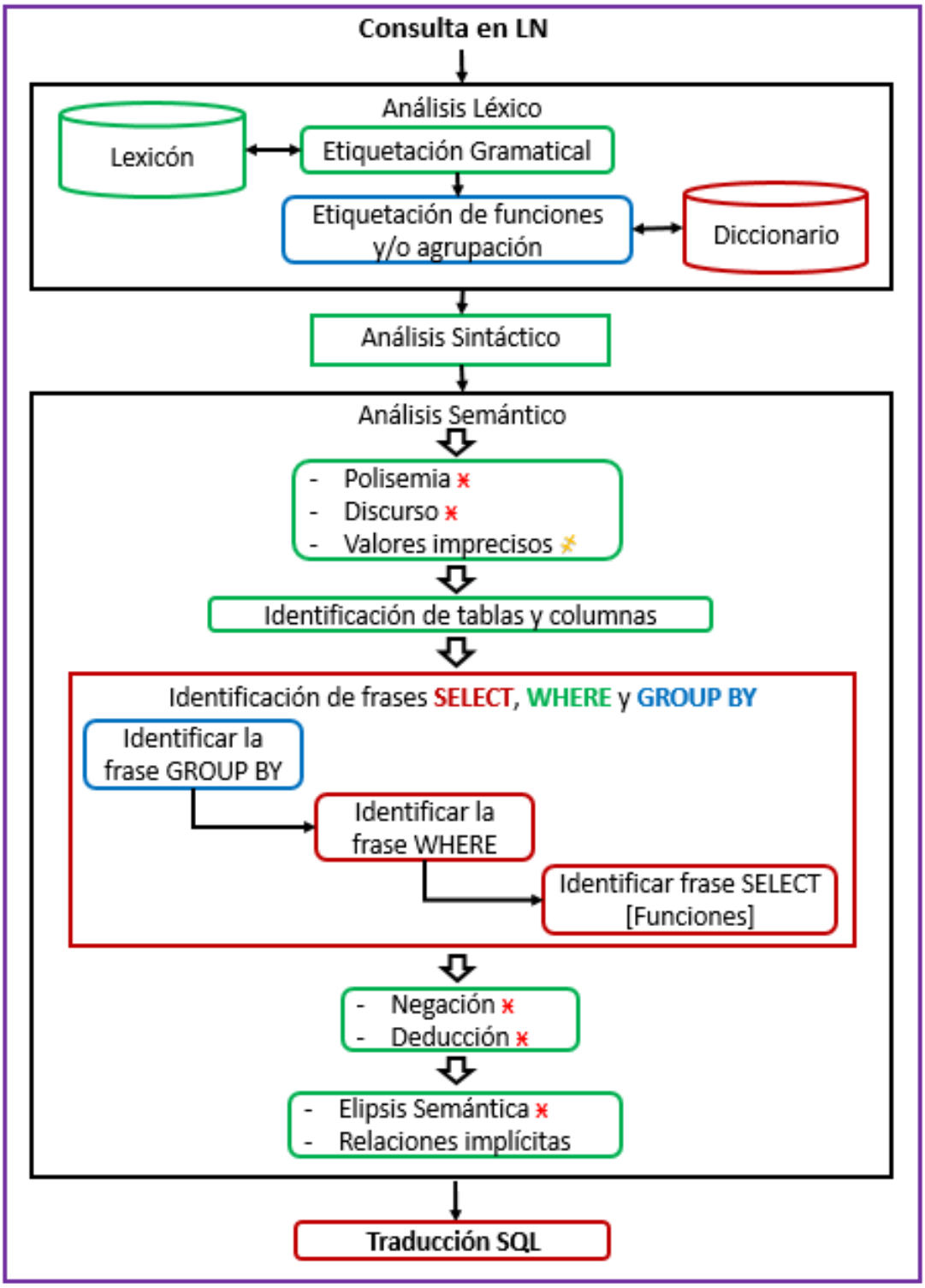


Figura 4.1. Diagrama general actualizado

Capítulo 5

Implementación del traductor

Este capítulo muestra los procedimientos realizados para la programación del diseño presentado en el capítulo anterior.

5.1. Actualización del Análisis Léxico

La primera fase de la traducción de las consultas de LN a SQL es el análisis léxico, en el cual se identifican los componentes de la consulta. El análisis léxico involucra el uso de un lexicón del idioma español y también un diccionario de datos de la BD que se consulta (ver Figura 5.1).

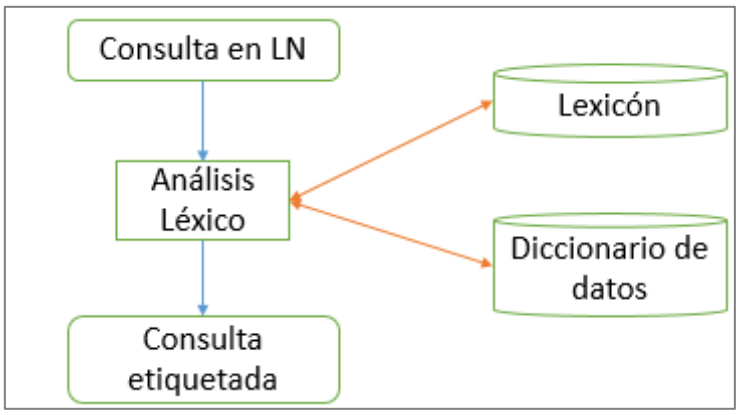


Figura 5.1. Análisis léxico del traductor

El diccionario de datos de la Interfaz contiene la información correspondiente a todos los elementos de la BD (tablas, columnas, relaciones, etc.), cada uno de estos elementos se identifica con un descriptor nominal en el idioma español.

Se actualizó el diccionario de datos para que el traductor identifique elementos de una consulta correspondientes a funciones de agregación y agrupamiento. Se agregó una tabla nombrada ***funciones_de_agregacion_agrupamiento*** con dos columnas (***palabra_frase*** y ***funcion_clausula***) en las cuales se almacenan las palabras y/o frases correspondientes a funciones de agregación y agrupamiento encontradas en el análisis realizado.

La etiquetación de las palabras y frases como FA o agrupamiento se realiza después del etiquetado gramatical de cada componente de la consulta en LN. Un ejemplo de lo anterior se muestra en la Tabla 5.1.

Tabla 5.1. Análisis léxico de una consulta en LN

Muestra	el	**número**	**de**	ciudades	**en**	**cada**	estado
ver	art	sus	pre	sus	pre	adj	sus
		funcion			**agrupamiento**		

El seudocódigo utilizado para actualizar el módulo de análisis léxico se presenta en la figura 5.2.

```
Procedimiento de Análisis Léxico
   Consulta ← ConsultaEnLN

   //Primera fase – etiquetación gramatical//
   Para Cada token ∈ consulta Hacer
            token.etiqueta = buscarEnLexicon (token);
   Fin Para Cada

   //Segunda fase – identificar funciones y agrupamiento
   Para Cada token ∈ consulta Hacer
            // Las frases se conforman de 2 componentes//
            Si posición (token) >= 2 Entonces
                     Si estaEnDiccionario (token_anterior + token) Entonces
                              token.etiqueta = [funcion or agrupamiento];
           Continuar…
                     Fin Si
            Si no Entonces
                     Si estaEnDiccionario (token) Entonces
                              token.etiqueta = [funcion or agrupamiento];
                     Fin Si
            Fin Si
   Fin Para Cada
Fin de Procedimiento
```

Figura 5.2. Algoritmo añadido al análisis léxico

5.2. Actualización del Análisis Semántico

El esquema del análisis semántico propuesto por Aguirre [1] y que se actualizó para en este proyecto se muestra en la figura 5.3, en el cual se marca de color rojo los módulos actualizados. Los módulos en color azul representan procesos que se agregaron para traducir consultas que involucran FAs y/o agrupamiento.

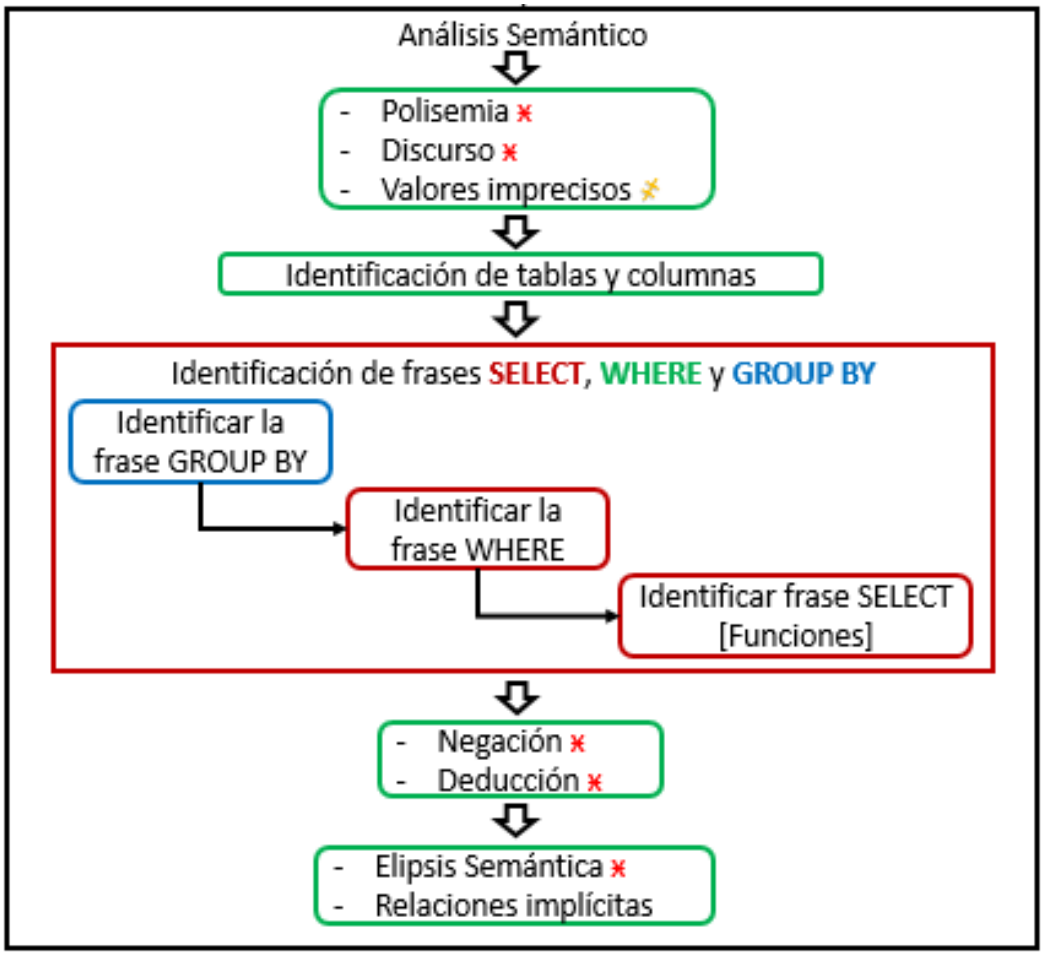


Figura 5.3. Esquema del análisis semántico actualizado

Las actualizaciones realizadas al análisis semántico se centran principalmente en el submódulo *identificación de la frase SELECT y WHERE*, en el cual se añadió la capacidad de identificar la cláusula GROUP BY. Este submódulo es subsiguiente al submódulo "***identificación de tablas y columnas***", en el cual se identifica cada uno de elementos de la consulta y se determina si es una tabla o una columna de la BD; según el diccionario de datos de la BD.

Un ejemplo del funcionamiento del módulo ***identificación de tablas y columnas*** se presenta en la tabla 5.2.

Tabla 5.2. Procesamiento (*identificación de tablas y columnas*)

Muestra	el	**número**	**de**	**ciudades**	**en**	**cada**	**estado**
ver	art	sus	pre	sus	pre	adj	sus
		funcion			**agrupamiento**		
				columna			**columna**

5.2.1. Identificación de frases SELECT, WHERE y GROUP BY

Este submódulo identifica cada una de las frases que conforman una consulta a su equivalente a SQL (SELECT, WHERE y GROUP BY). Este submódulo consta de 3 fases de las cuales 2 se actualizaron. A continuación de describen las fases que se actualizaron.

- **Identificación de frase GROUP BY**

La modificación a este submódulo consistió en reconocer la frase GROUP BY de una consulta, para lo cual y con base al análisis realizado sobre este tipo de consultas, se pudo observar que el agrupamiento se indica de la misma forma en todas las consultas analizadas. A continuación se muestra un ejemplo.

Consulta en LN: *Muestra el número de ciudades* ***en cada*** *estado*

En la consulta anterior que involucra agrupamiento, la frase reservada "***en cada***" indica que se debe utilizar la cláusula GROUP BY para traducir correctamente la consulta, y el elemento siguiente (*estado*), es el atributo al cual se le aplica dicha condición de agrupamiento.

A partir de la característica anterior, el traductor para reconocer la frase GROUP BY busca todas las palabras y frases reservadas para reconocer el agrupamiento y la relaciona al elemento siguiente inmediato. Al identificar estos elementos la consulta se representa como se muestra en la tabla 5.3.

Tabla 5.3. Procesamiento (*Identificación de la frase GROUP BY*)

Muestra	el	**número**	**de**	**ciudades**	**en**	**cada**	**estado**
Ver	art	sus	pre	sus	pre	adj	sus
		funcion			**agrupamiento**		
				columna			**columna**
					GROUP BY (estado)		
					Frase GROUP BY		

Al finalizar el procesamiento de este submódulo se continúa el procesamiento de la consulta en el submódulo "***Identificación de la frase WHERE***", el cual se realiza según lo expuesto por [Aguirre] y de acuerdo al diseño presentado anteriormente no se realiza ninguna alteración de este submódulo en el presente proyecto.

- **Identificación de frase SELECT**

La actualización de esta fase consistió principalmente en encontrar los elementos de la(s) función(es) de agregación que se encuentra(n) en las consultas e incorporarlos a la frase SELECT.

La identificación de los elementos de las funciones de agregación se realizó siguiendo los patrones que se encontraron en nuestro análisis y que se presentan en la tabla 4.3. A continuación se muestra un ejemplo de la identificación de elementos de las funciones de agregación.

Consulta en LN: Muestra el número de ciudades en cada estado.

En la consulta la frase "***número de***" hace referencia a una función de agregación (COUNT), pero aún no se sabe a qué elemento hace referencia esta función. Para encontrar el elemento indicado se realiza una búsqueda de los elementos de la consulta que sean columna y que se encuentren de acuerdo a los patrones de la tabla 4.3. Para este ejemplo el patrón asociado es el número 2. El proceso de traducción hasta este paso queda como se muestra en la tabla 5.4.

Tabla 5.4. Procesamiento (*Identificación de la frase SELECT*)

Muestra	el	**número**	**de**	**ciudades**	**en**	**cada**	**estado**
ver	art	sus	pre	sus	pre	adj	sus
		funcion			**agrupamiento**		
				columna			**columna**
					GROUP BY (estado)		
					Frase GROUP BY		
		COUNT (ciudades)					
		Frase SELECT					

El seudocódigo del algoritmo que permite realizar el procesamiento de la consulta en este módulo se muestra en la figura 5.4.

```
Procedimiento de Análisis Semántico
Consulta ← resultadoDeAnalisisSintactico
// Identificar frase GROUP BY //
  Para Cada agrupamiento ϵ consulta Hacer
// Las columnas relacionadas con el agrupamiento siempre aparecen a la derecha //
    Si ExisteColumnaALaDerecha Entonces
            Relacionar (agrupamiento, columna);
        Si No Entonces
            DialogoParaAclararConsulta
        Fin Si
  Fin Para Cada

  // Identificar frase SELECT //
        Para Cada funcion ϵ consulta Hacer
        // Las columnas relacionadas con las funciones pueden aparecer en 5 patrones** //
    Si obtenerColumnaDeFA (funcion) Entonces
        Relacionar (funcion, columna)
    Si no Entonces
            DialogoParaAclararConsulta
    Fin Si
  Fin Para Cada
Fin de Procedimiento
```

Figura 5.4. Seudocódigo añadido al análisis semántico

5.3. Actualización del proceso de traducción a SQL

El módulo Traducción SQL genera la consulta SQL final, la cual es posteriormente ejecutada sobre la BD a consultar.

En este módulo se definen los elementos de la consulta a una sentencia del lenguaje SQL. La actualización realizada en este módulo permitió definir elementos SQL para consultas que involucran FAs y/o agrupamiento.

El funcionamiento de este módulo en la versión de Aguirre consistía de 3 etapas, las cuales se mencionan a continuación:

1. **Generar la cláusula WHERE**: A partir de la información obtenida en el análisis de la consulta, se identifican aquellos elementos (condiciones) que formarán la frase WHERE y se añaden a la consulta SQL.
2. **Generar la cláusula SELECT**: Después de generar la cláusula WHERE se identifican los elementos que corresponderán a la frase SELECT, éstos se agregan a la instrucción SQL, además, esta etapa se actualizó para definir los elementos asociados a la(s) función(es) de agregación que presenta la consulta.
3. **Generar la cláusula FROM**: La cláusula FROM se genera a partir de las cláusulas SELECT y WHERE, consiste en identificar las tablas involucradas en dichas cláusula e incorporarlas a la consulta SQL.

Para que la consulta SQL incluya la cláusula GROUP BY se añadió la siguiente etapa.

4. **Generar la cláusula GROUP BY**: Se identifican los elementos marcados correspondientes a la frase GROUP BY y se incorporan a la consulta SQL.

También se añadió una etapa que permite definir los elementos asociados a una sub-consulta en SQL.

5. **Generar subconsultas**: Se identifican y se incorporan a la consulta SQL los elementos que se han marcado como subconsulta en el análisis de la consulta en los módulos anteriores. Las subconsultas pueden ser de dos tipos:

    I. Función de agregación en cláusula WHERE.

    Ejemplo:

    LN: *Muéstrame el río de mayor longitud*

    SQL:
    ```
    SELECT river.name
    FROM   river
    ```

```
WHERE  river.length = ( SELECT      MAX (river.length)
                        FROM        river );
```

II. Función de agregación y condición en subconsulta

Ejemplo:

LN: *Dame la ciudad con más población en el estado de Ohio*

SQL:
```
SELECT    city.name
          FROM      city
          WHERE     city.population = (SELECT          MAX
(city.population)
                    FROM              city, state
                    WHERE             state.id = city.state
                                      AND state.name like 'ohio');
```

Capítulo 6

Resultados

Este capítulo presenta las pruebas realizadas al traductor después de la implementación descrita en el capítulo anterior. Se muestran los resultados obtenidos de las diferentes pruebas realizadas.

6.1. Pruebas

Las pruebas que se realizaron al traductor consistieron en ejecutar cada una de las consultas de los corpus analizados y comprobar si el traductor realizaba la traducción correcta en SQL.

El entorno de pruebas consistió en una aplicación Web a la cual se incorporó el traductor con el fin de realizar consultas de manera remota al traductor.

El diccionario de datos utilizado fue generado por el traductor en el proceso de configuración. El traductor se implementó en el lenguaje Java y su utilizó Apache Tomcat como servidor de aplicación para realizar las pruebas.

La figura 6.1 muestra la página principal de la aplicación utilizada para realizar las pruebas al traductor.

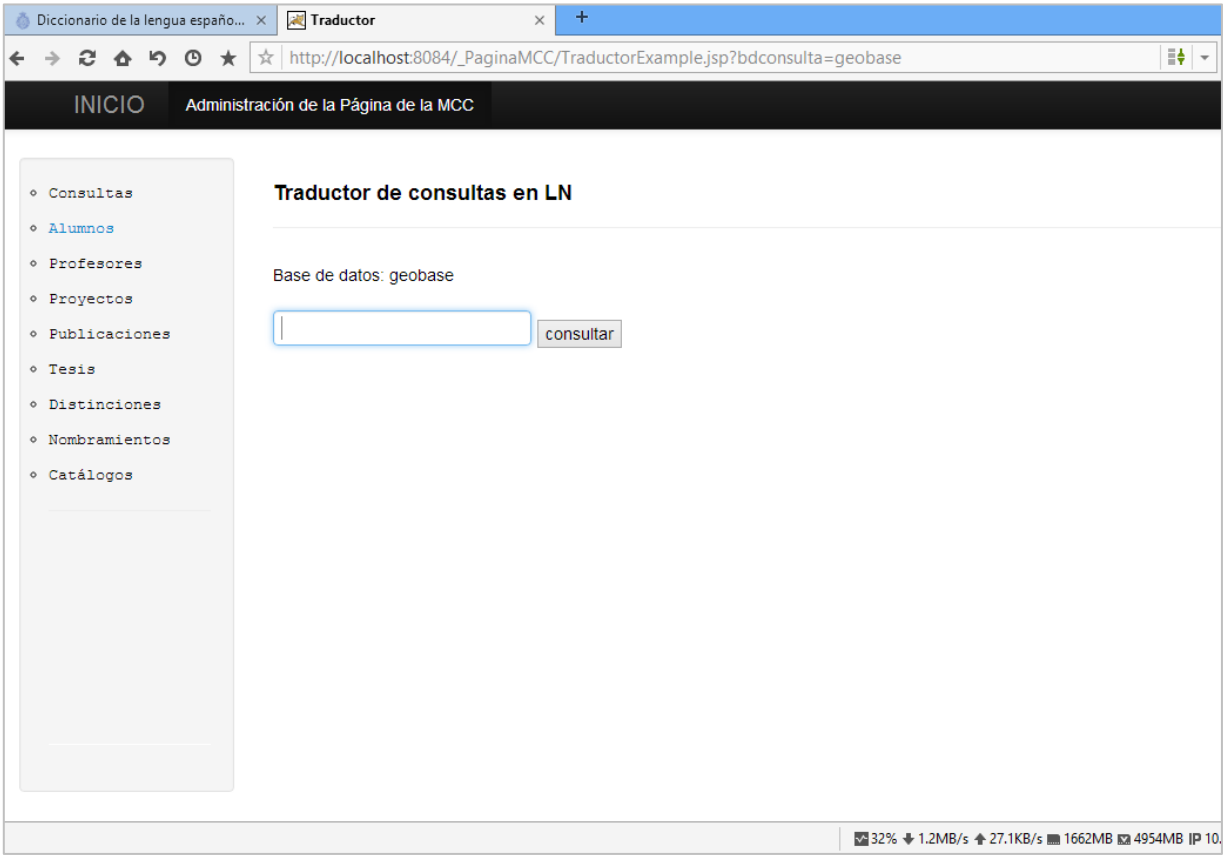


Figura 6.1. Aplicación Web para realizar pruebas al traductor

En las secciones siguientes se muestran ejemplos de las traducciones realizadas por el traductor y los resultados obtenidos en las pruebas.

6.2. Pruebas con los Corpus Analizados

La primera prueba se realizó utilizando los corpus que se estudiaron en el análisis de este trabajo. Esta prueba consistió en introducir a la ILNBD las consultas que involucran alguna función de agregación y/o agrupamiento y cumplan con las características especificadas en los alcances y limitaciones del proyecto. La tabla 6.1 muestra algunos ejemplos de consultas que involucran FAs y/o agrupamiento.

Tabla 6.1. Consultas con FA y/o agrupamiento

#	**Consulta**	**BD**
1	¿Cuál es el río con la mayor longitud?	GEOBASE
2	¿Cuántas ciudades hay en cada estado?	GEOBASE
3	¿Cuántas materias se tienen registradas?	ITCM-MCC
4	¿Dame la mayor área de los lagos?	GEOBASE
5	¿Cuántos alumnos hay por carrera?	ITCM-MCC

Los resultados de la prueba descrita para los corpus de la BD ITCM-MCC y GEOBASE se presentan en la tabla 6.2, como se puede observar el desempeño del traductor se encuentra por encima de 80%. La tabla 6.3 muestra el resultado del resto de los corpus, debido al número pequeño de consultas con características de este proyecto, no se describe el porcentaje de desempeño.

Tabla 6.2. Resultados obtenidos con los corpus de las BDs ITCM-MCC y GEOBASE.

Corpus	**Consultas**	**Correctas**	**Porcentaje**
ITCM-MCC	87	73	**83%**
Geobase	82	74	**90%**

Tabla 6.3. Resultados de los corpus de las BDs PUBS, NORTHWIND y GMISARA.

Corpus	Consultas	Correctas
PUBS	7	7
NORTHWIND	1	1
GMISARA	3	2

6.3. Pruebas en Ambiente Real

Esta prueba consistió en realizar consultas al traductor de manera concurrente, para ello se organizó un grupo de estudiantes de licenciatura a los cuales se les proporcionó un manual para entender la BD (GEOBASE) a la que formularon consultas. El manual contenía el diagrama Entidad-Relación, el diccionario de datos y ejemplos de consultas a la BD.

El grupo de estudiantes ejecutó consultas a la ILNBD de manera libre y anotaba ***correcta*** si la respuesta de la interfaz lo era, de lo contrario se anotaba como ***incorrecta***. La tabla 6.4 muestra los resultados obtenidos en esta prueba.

Tabla 6.4. Resultados de la prueba en ambiente real

BD	Consultas	Correctas	Porcentaje
GEOBASE	255	237	92%

Capítulo 7

Conclusiones y Trabajos Futuros

Las interfaces de lenguaje natural para bases de datos brindan la posibilidad de obtener información de una manera sencilla para los usuarios inexpertos en el área de bases de datos. Estas herramientas dan oportunidad para el desarrollo de nuevas tecnologías y proporcionan alternativas para la recuperación de información.

La traducción de consultas que involucran FA(s) y/o agrupamiento es un proceso difícil por lo cual la mayoría de las ILNBD que se encontraron en la literatura no la toman en cuenta. En la práctica este tipo de consultas se han empezado a utilizar con más frecuencia, lo cual motivo a realizar este proyecto. Los resultados obtenidos con la ILNBD desarrollada en el ITCM-CENIDET muestran que la traducción de consultas que involucran agrupamiento se ha logrado realizar de manera satisfactoria.

Los resultados obtenidos por el traductor después de realizar pruebas con consultas de los corpus analizados muestran una efectividad de entre el 83% y 92%. Entendiéndose como efectividad el porcentaje de consultas correctas con respecto al número de consultas que se ejecutaron.

7.1. Trabajos Futuros

El desarrollo de este proyecto permite continuar mejorando la interfaz de LN a BD desarrollada en el ITCM, aunque faltan muchas necesidades para lograr la traducción correcta de cualquier consulta de LN a SQL. Para continuar con el desarrollo de la ILNBD es necesario el desarrollo de procesos de diálogo para el tratamiento de problemas que presentan las consultas que involucran alguna FA y/o agrupamiento, también es necesario el desarrollo de un sistema de aprendizaje para mejorar los resultados en consultas que involucran FA.

Bibliografía

[1] Pazos R., Gonzales y Aguirre L, «Semantic Model for Improving the performance of Natural Language Interfaces,» *10th Mexican International Conference on Artificial Intelligence,* pp. 277-290, 2011.

[2] Peter Haggar, Senior Software Engineer, IBM, «Crecimiento de datos y estándares,» 08 08 2011. [En línea]. Available: http://www.ibm.com/developerworks/ssa/xml/library/x-datagrowth/.

[3] J. C. Rojas, Administrador de dialogo para una interfaz de Lenguaje Natural para Base de Datos, Cuernava, México, 2009.

[4] J. J. G. Barbosa, Traductor de Lenguaje Natural Español a SQL para un sistema de Consultas a Bases de Datos, Cuernavaca, México., 2005.

[5] R. Pazos R. y J. Pérez, «A Domain Independent Natural Language,» *Lecture Notes in Artificial Intelligence,* pp. 203-213, 2005.

[6] M. A. Aguirre Lam, «Interfaces de Lenguaje Natural para Consultar Bases de Datos en Español,» *Komputer Sapiens,* pp. 20-24, 2012.

[7] R. A. Española, «rae.es,» 11 Octubre 2012. [En línea]. Available: http://lema.rae.es/drae/?val=lenguaje.

[8] E. G. Díaz, «Monografias.com,» 2004. [En línea]. Available: http://www.monografias.com/trabajos17/lenguaje-natural/lenguaje-natural.shtml. [Último acceso: 21 Enero 2014].

[9] F. -. F. O. D. o. Computing, «The Free Dictionary,» [En línea]. Available: http://encyclopedia2.thefreedictionary.com/natural+language. [Último acceso: 21 Enero 2014].

[10] D. Liddy, *Natural Language Processing for information retrieval & knowledge Discovery,* 2001.

[11] Elmasri, Fundamentos de Sistemas de Bases de Datos, 5ta Edición: Pearson, 2007.

[12] I. Androutsopoulos, G. Ritchie y P. Thanisch, *MASQUE/SQL - An Efficient and Portable Natural Language Query Interface for Relational Databases,* 1993.

[13] S. K. y. S. Silberschatz, *Fundamentos de Bases de Datos,* McGraw- Hill, 2008.

[14] Y. L. F. W. L. S. K. F. Vincent, «A Query Interface Truly for Chinese Users,» *Department of Systems Engineering and Engineering Management,* 1995.

[15] S. M. Idicula, Design and development of an adaptable Frame-Based System for Dravidian

Language Processing, COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY, 1999.

[16] M. ,. Z. B. K. C. Owda, «Conversation-Based Natural Language Interface to Relational Databases,» *The Intelligent Systems Group, Department of Computing and Mathematics,The Manchester Metropolitan University, Chester Street, Manchester, M1 5GD, UK,* 2007.

[17] Y. I. C. H. Y. S. S. H. V. J. Li, *DaNaLIX: a Domain-adaptive Natural Language Interface for Querying XML,* 2007.

[18] R. A. Bhootra, Natural Language Interfaces: Comparing English Language Front End and English Query, Richmond, Virginia, 2004.

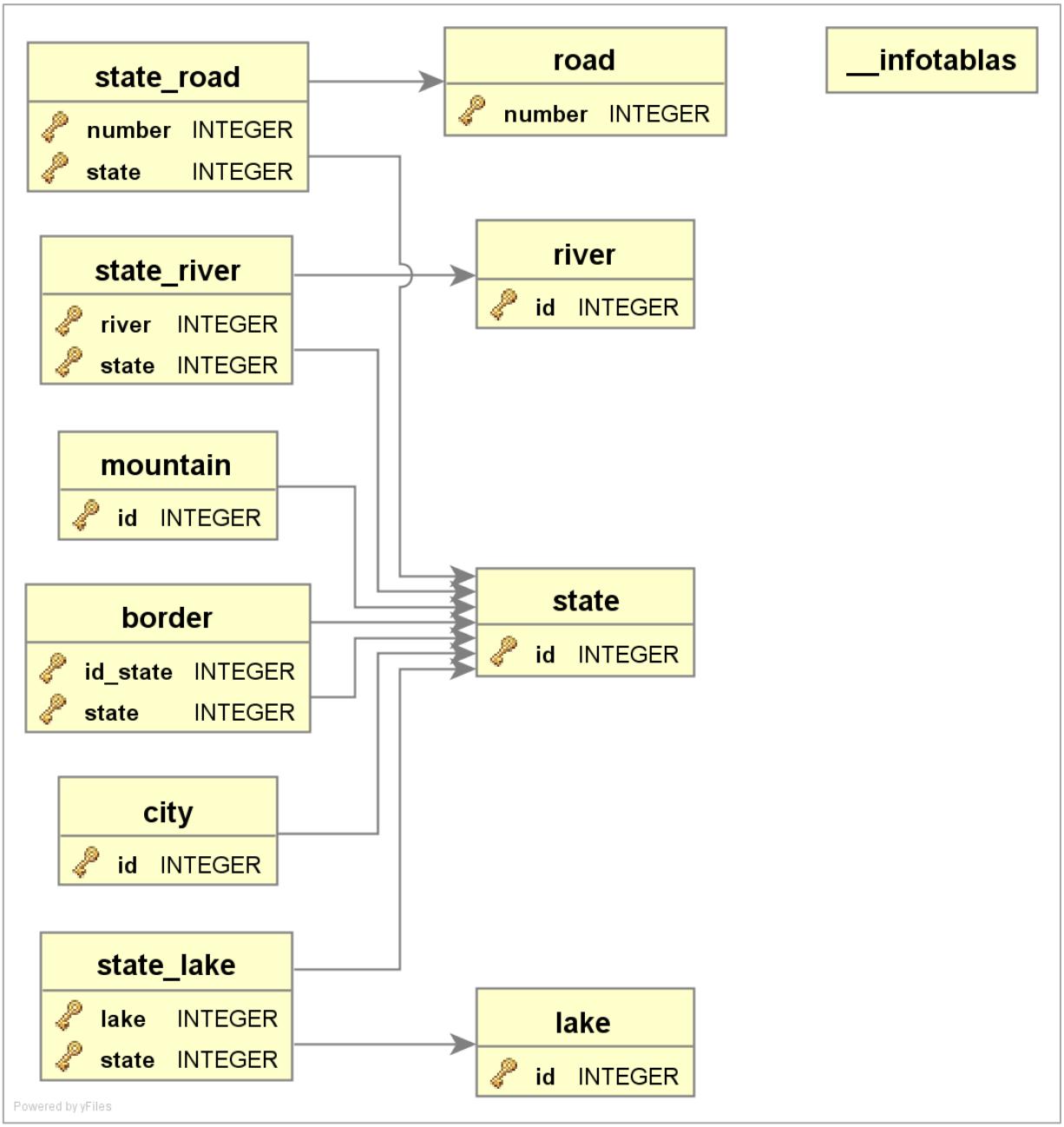
[19] I. M. ;. M. d. l. Á. G. Esquivel, «Translation of Spanish Statistics Expressions to SQL,» *Advances in Soft Computing Algorithms. Research in Computing Science ,* pp. 39-46, 2010.

[20] A. Copestake y K. Sparck Jones, *Natural Language Interfaces to Databases,* 1989.

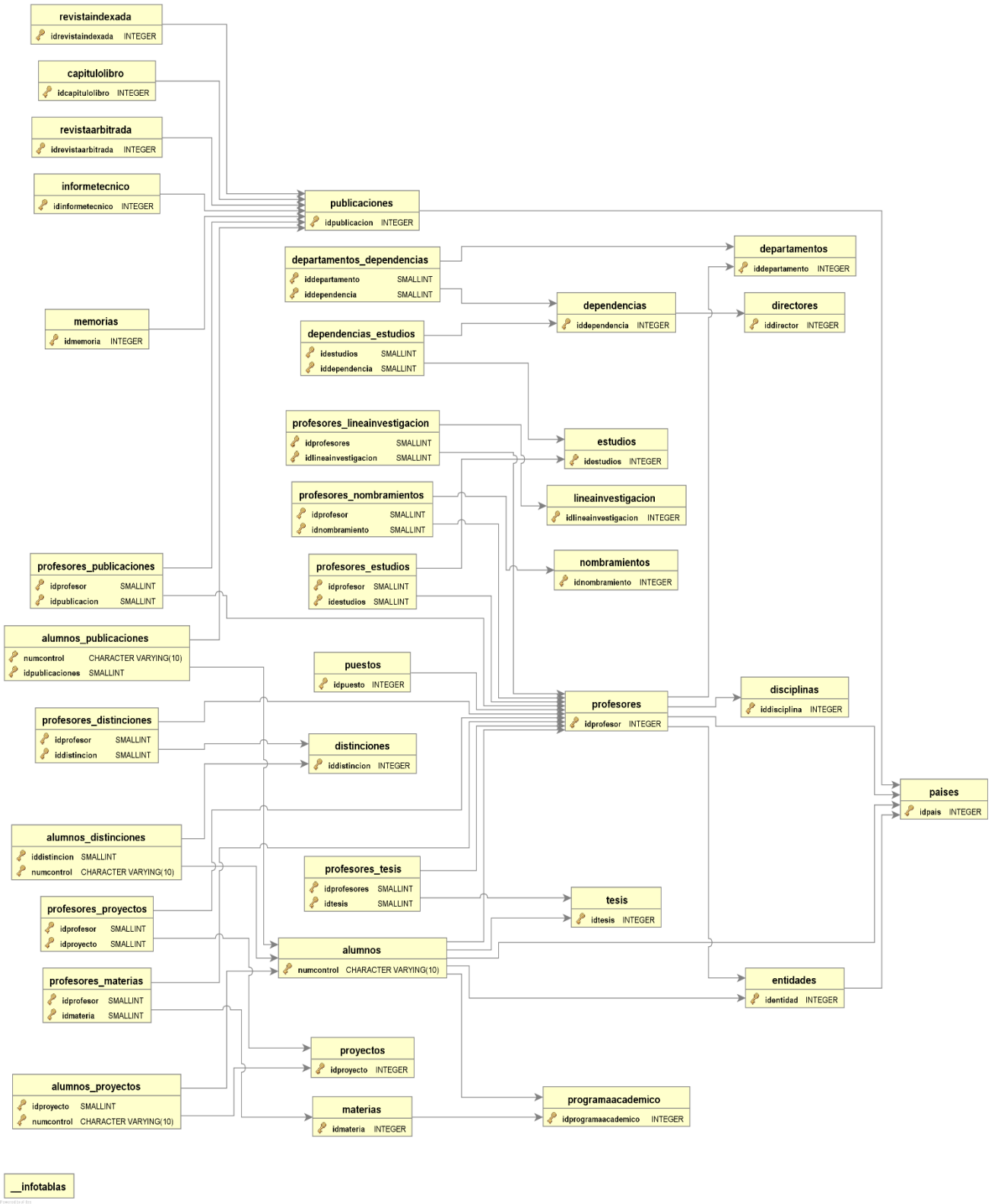
[21] I. Androutsopoulos, «Interfacing a Natural Language Front-End to a Relational Database,» 1992.

Anexos

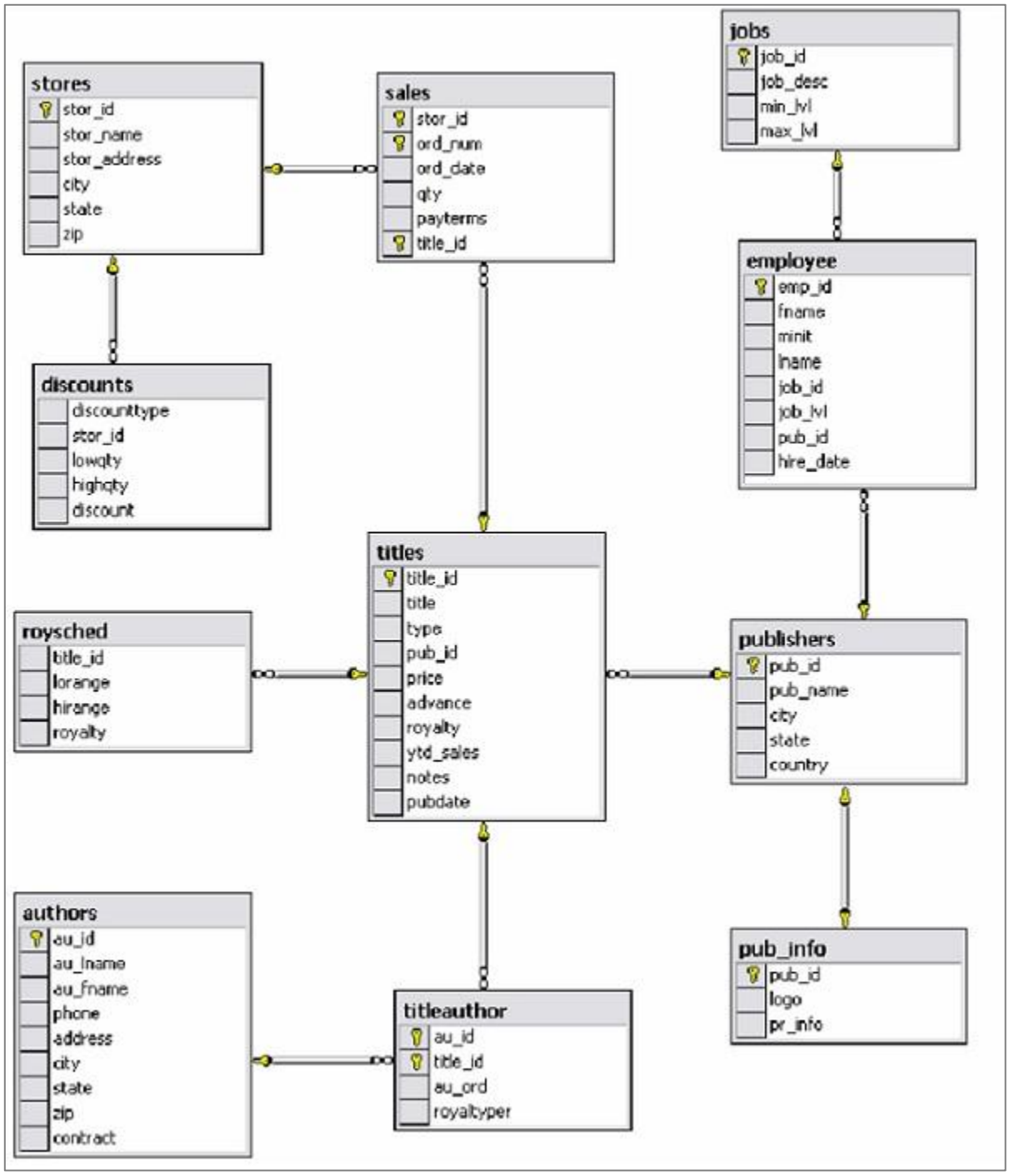
Anexo 1. Diagrama de relaciones de la BD Geobase



Anexo 2. Diagrama de relaciones de la BD ITCM-MCC



Anexo 3. Esquema de la BD Pubs



Anexo 4. Esquema de la BD Northwind