

# Una Herramienta para el Aprendizaje del Álgebra Relacional

Carmen Hernández, Yania Crespo, Pilar Romay, Miguel Angel Laguna

Departamento de Informática

Universidad de Valladolid

47011 Valladolid

e-mail: {chernan, yania, mpromay, mlaguna}@infor.uva.es

## Resumen

El estudio de las bases de datos relacionales forma parte del currículum de los estudios de Ingeniería Técnica Informática. Los lenguajes de consulta abstractos, entre ellos el Álgebra Relacional, forman una parte importante de este estudio, pero nuestra experiencia docente nos ha demostrado que, para los alumnos, es difícil conocer cuándo las consultas expresadas en un papel en términos de estos lenguajes son correctas y responden a los requisitos de información planteados.

En este trabajo se describe una herramienta de apoyo que se ha desarrollado en la Universidad de Valladolid que permite realizar consultas expresadas en Álgebra Relacional sobre cualquier base de datos. El alumno puede explorar, así, las diferentes posibilidades de este lenguaje abstracto, comprobando por sí mismo la calidad de su aprendizaje. La herramienta ha sido desarrollada siguiendo guías metodológicas propugnadas en el ámbito del diseño de entornos de aprendizaje.

## 1. Introducción

El modelo relacional se ha establecido como el principal modelo de datos para la construcción de Sistemas de Información. Este modelo tiene unos sólidos fundamentos matemáticos, ya que se basa en la Teoría de Conjuntos. Esta base teórica ayuda al diseño de las bases de datos relacionales y, particularmente, al procesamiento eficiente de las peticiones de información de los usuarios. El modelo define de forma precisa los diferentes lenguajes abstractos con los que el usuario solicita información de la base de datos. Uno de estos lenguajes es el Álgebra Relacional.

El estudio de las bases de datos relacionales forma parte del currículum de los estudios de Ingeniería Técnica Informática desde la implantación de éstos en la Universidad de Valladolid. En particular, los lenguajes de consulta abstractos forman una parte importante de este estudio. Sin embargo, nuestra experiencia docente nos ha demostrado que para los alumnos es difícil conocer cuándo las consultas expresadas en el papel en términos del Álgebra Relacional son correctas y responden a los requisitos de información planteados.

Parecía de especial interés, por tanto, desarrollar una herramienta que proporcionara un mecanismo mediante el cual los alumnos puedan explorar las diferentes posibilidades de estos lenguajes. Además, se debería permitir una realimentación inmediata, de manera que vieran el resultado de la consulta que habían propuesto en ese mismo momento y pudieran realizar las oportunas modificaciones.

La herramienta que hemos construido pone especial cuidado en conseguir la pretensión inicial de facilitar el aprendizaje. Hemos revisado algunas de las teorías existentes sobre el diseño de entornos de aprendizaje, intentando concretar las principales características de este tipo de entornos y las guías existentes para la construcción de los mismos.

En la siguiente sección se presenta el entorno educativo en el que se ha desarrollado la herramienta. La tercera sección sirve para introducir brevemente algunos conceptos fundamentales del diseño de entornos de aprendizaje. En la cuarta sección presentamos la herramienta que hemos desarrollado. Finalmente, exponemos las líneas de nuestro trabajo futuro y las conclusiones de éste.

## 2. Entorno educativo

El modelo relacional define de forma precisa los diferentes lenguajes abstractos con los que un usuario solicita información de la base de datos. Estos lenguajes de consulta pueden clasificarse en procedimentales y no procedimentales. En los procedimentales, el usuario indica al Sistema Gestor de Base de Datos Relacional (SGBDR) la serie de operaciones que ha de realizar en la base de datos para obtener el resultado deseado. En los no procedimentales el usuario describe la información deseada sin dar un procedimiento concreto para obtener esa información.

La mayor parte de los SGBDR ofrecen un lenguaje de consulta que incluye elementos de los enfoques procedimental y no procedimental. El lenguaje SQL (Structured Query Language) es un ejemplo de lenguaje que se ajusta a los dos enfoques.

Los lenguajes abstractos ilustran de forma precisa las técnicas fundamentales para la extracción de datos de las bases de datos, obviando los aspectos sintácticos de los lenguajes comerciales. El Álgebra Relacional es un lenguaje abstracto de consulta procedimental. Consta de un conjunto de operaciones que toman como entrada una o dos relaciones y producen como resultado una nueva relación.

Como ya se ha comentado, el estudio de las bases de datos relacionales forma parte del currículum de los estudios de Ingeniería Técnica Informática. Los lenguajes de consulta abstractos, fundamentalmente el Álgebra Relacional, forman una parte importante de este currículum.

Nuestra hipótesis de trabajo es que los alumnos aprenden más rápidamente el lenguaje SQL cuando ya están familiarizados con las cuestiones que subyacen en los lenguajes de consulta formales, mediante el uso del Álgebra Relacional [3]. En particular, la sentencia *select-from-where* de SQL se define precisamente en términos del Álgebra, de manera que el mecanismo mediante el cual el SGBDR es capaz de recuperar información surge ante el alumno de una manera casi espontánea.

Nuestra experiencia, tras varios años impartiendo esta asignatura, nos permite asegurar que para los alumnos es difícil conocer cuándo una consulta expresada en Álgebra y escrita en un

papel es correcta. Esta dificultad se extiende al profesor, al tener que controlar la corrección de algunas consultas “creativas” que a veces se proponen.

Todo esto justifica la necesidad de utilizar una herramienta que proporcione un mecanismo mediante el cual los alumnos puedan explorar las diferentes posibilidades de estos lenguajes, permitiendo, además, una realimentación inmediata, de manera que ellos vean el resultado de la consulta que habían propuesto en ese mismo momento y puedan realizar las oportunas modificaciones, hasta alcanzar una respuesta satisfactoria.

Durante algunos años hemos estado utilizando en nuestro laboratorio de Bases de Datos un prototipo de una herramienta que ha sido desarrollada en Arizona State University de EE.UU. [2], que también tiene como objetivo el aprendizaje de estos lenguajes.

Este prototipo no se ajusta estrictamente a los requisitos que nosotros tenemos planteados, ya que no tiene incorporados todos los operadores que presentamos en la asignatura, con lo que algunas de las cuestiones que planteamos al alumno no tienen respuesta en el ámbito de esta herramienta.

Con esta experiencia y con las lecciones que hemos aprendido en nuestra práctica docente, hemos desarrollado nuestra propia herramienta, que se ajusta exactamente a los contenidos de nuestra asignatura, de modo que, realmente, se facilite el aprendizaje del alumno.

## 3. Diseño de entornos de aprendizaje

Todo proceso de aprendizaje está condicionado por las variables del medio en que se desarrolla, de manera que este proceso está determinado por el entorno en el que tiene lugar. Cuando se trata de un proceso condicionado por un determinado software, el aprendizaje es estimulado y dirigido por aspectos diversos que vienen condicionados por el nivel del producto alcanzado, por su riqueza cualitativa y por su complejidad.

Podemos entender el entorno en el que se desarrolla el aprendizaje como un marco de situaciones y actividades que orientan y guían el aprendizaje, lo condicionan y lo estimulan para que tome una dirección u otra. En este sentido, el

entorno lleva de la mano al usuario por los caminos previamente fijados por el diseñador, ya que cualquiera que sea la propuesta que se presente al usuario, abierta o cerrada, o con múltiples opciones, todas han sido previamente definidas en función de la idea de guiar el proceso [6].

Se podría decir, entonces, que lo que el diseñador hace con su trabajo es una anticipación del proceso de aprendizaje que realizará el usuario. El diseñador realiza una actividad de enseñanza al organizar los ambientes y los elementos que orientarán el aprendizaje de los usuarios. La tarea de enseñar implica siempre, en cualquier contexto, ya sea virtual o presencial, el diseño de entornos y la previsión de situaciones cuya intención es que conduzcan al éxito en el aprendizaje.

Teniendo todo esto en cuenta, el diseño de software educativo ha de ser concebido como una construcción metodológica. Esto significa que no es absoluta, ni para todas las situaciones, ni para todos los contenidos e individuos y que, por lo tanto, debe ser planteada, en el marco de situaciones concretas, en un contexto determinado [5]. De esta manera, se puede hablar del diseño como un proceso que toma la forma de espiral constructivo, y no como una secuencia lineal de selección de los elementos que lo integran. Todo esto determina el ciclo de vida que mejor se ajusta al tipo de sistema que pretendemos construir.

En el diseño de este tipo de sistemas intervienen cuatro elementos básicos a partir de los cuales se determina el material a construir: la estructuración del contenido, las actividades a realizar, los recursos con que se cuenta y las formas de interacción que se prevén, generando en conjunto un ambiente o entorno que ayude u oriente el aprendizaje del usuario [4].

La secuencia y organización de los contenidos determina las formas de apropiación de los conocimientos por parte de los usuarios. Esto hace que la selección de estos sea una etapa crucial que demanda del diseñador un profundo conocimiento de las disciplinas científicas que están en juego en lo referente a los conceptos principales que se quieren transmitir.

Otra etapa fundamental en este planteamiento es la estructuración de las actividades que van a desarrollar los usuarios para operar con el sistema. Las actividades que se propongan deben estimular

la actitud de interrogación. El usuario debería visualizar las consecuencias de su acción sobre el problema, elaborar hipótesis y comprobarlas durante el proceso, visualizando los resultados de sus deducciones, a través de la experimentación.

Planear la actividad implica generar un proceso reflexivo y de construcción del conocimiento por parte del diseñador, que debe enfrentarse al mismo desafío que se planteará el usuario.

Como consecuencia de todo lo expuesto, hemos afrontado la tarea de diseñar esta herramienta mediante un ciclo de vida en espiral [1], de forma que nos encontramos delante del primer prototipo de esta herramienta en el que se han tenido en cuenta, fundamentalmente, la estructuración de los contenidos y los recursos que se van a necesitar.

El equipo de diseñadores ha estado formado por profesores involucrados en las diferentes asignaturas de Bases de Datos que se imparten y por alumnos que realizaban su Proyecto Fin de Carrera. Los primeros se han encargado de proporcionar el conocimiento de la disciplina que se intentaba dar a conocer y los segundos han asumido perfectamente el papel del usuario que debe enfrentarse al problema de aprender un lenguaje de consulta nuevo, como es el Álgebra Relacional.

#### 4. La herramienta

La herramienta que se ha desarrollado [7] pretende facilitar la comprensión de las siguientes materias:

- Operadores que constituyen el Álgebra Relacional.
- Combinaciones de operadores para formar consultas.
- Semántica del Álgebra Relacional.
- Sintaxis y semántica del SQL.

Según lo expuesto en el apartado anterior, para que la herramienta sea útil desde un punto de vista didáctico, los alumnos deben de poder emitir respuestas a las cuestiones que les hayan sido planteadas y el sistema tendrá que detectar los errores concretos que se hayan cometido, además de ofrecer la posibilidad de hacer modificaciones a lo respondido.

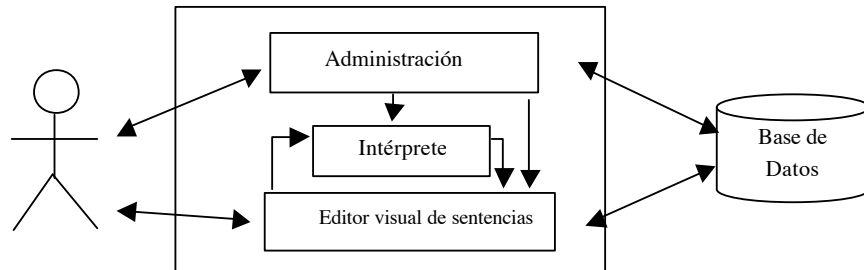


Figura 1. Arquitectura del sistema

Las funcionalidades básicas que la herramienta debe cubrir son:

- Mantener el esquema de la base de datos con la que se va a trabajar.
- Mantener la instancia de la base de datos.
- Definir consultas en Álgebra Relacional: Se introducirá una consulta mediante el teclado y el sistema la evaluará. Se debe dar la posibilidad de almacenar la consulta para utilizarla en otro momento.
- Visualizar resultados: El sistema evalúa la consulta y presenta el resultado en pantalla, pudiendo almacenarse la nueva tabla obtenida de forma permanente en la base de datos como una vista.

En todo momento se buscará la máxima interactividad con el alumno, de manera que el sistema siempre mantendrá informado al usuario de los errores que vaya cometiendo.

Para permitir la máxima disponibilidad de la herramienta, se ha desarrollado sobre una plataforma Windows, de forma que los alumnos puedan utilizarla en sus propios ordenadores.

#### 4.1. Arquitectura

La arquitectura, que se muestra en la Figura 1, favorece la disponibilidad y usabilidad de la herramienta desarrollada. En ella se pueden apreciar los distintos módulos que constituyen el sistema y los interfaces que se establecen entre ellos.

Se ha optado por dividir la funcionalidad del sistema construido en dos partes; por un lado se encuentra el proceso encargado de la traducción de la sentencia en Álgebra Relacional a un conjunto de instrucciones SQL equivalentes que

serán utilizadas para obtener los resultados de la consulta y, por otro, el proceso que actúa de interfaz con la base de datos y con el usuario.

Al dividir el sistema de este modo se puede afrontar la posterior implementación de cada uno de los procesos con las tecnologías más adecuadas para cada caso, teniendo en cuenta las funciones que deben cumplir.

El módulo *Intérprete* se encarga del proceso de traducción de las sentencias del Álgebra Relacional al conjunto de instrucciones SQL para *Access*, a partir de las cuales se obtendrá el resultado de la consulta equivalente a la sentencia inicial.

El módulo *Editor visual de sentencias* se encarga de la funcionalidad necesaria para crear, editar, compilar y guardar sentencias del Álgebra Relacional. Una vez obtenidas las instrucciones SQL para *Access* mediante el *Intérprete*, este módulo obtiene el resultado a través del motor de base de datos *Microsoft Jet 3.0* y lo visualiza.

El módulo *Administración* permite al usuario determinar la base de datos con la que va a trabajar. Obtiene la *ruta del origen de datos* y la estructura de la base de datos, que son almacenadas de forma apropiada para su posterior uso por el módulo *Editor visual de sentencias*.

La base de datos contiene los datos sobre los cuales va a trabajar el usuario a través de las sentencias del Álgebra Relacional.

En la implementación de la herramienta se ha optado por utilizar *Visual C++* en su versión 6.0 y la biblioteca *MFC (Microsoft Foundation Class)*. Además, se ha utilizado *Access* como base de datos y su motor de base de datos *Microsoft Jet* en su versión 3.0, lo que permite la comunicación con el sistema a través de las funciones de las clases *DAO (Data Access Objects)*. De esta

manera, el usuario podrá practicar con diferentes bases de datos, sin más que modificar el *origen de datos* con el que quiere trabajar.

#### 4.2. Interfaz de usuario

La primera vez que se utiliza la herramienta se advierte al usuario de la necesidad de configurar el *origen de datos*; es decir, la ruta de la base de datos sobre la que se va a trabajar. Esta base de datos debe existir, aunque posteriormente se podrá modificar su esquema añadiendo o eliminando nuevas tablas y actualizando las tuplas que pertenecen a cada una de ellas.



Figura 2. Operaciones implementadas.

Las operaciones que se han implementado en esta herramienta aparecen disponibles en una ventana flotante, de manera que el usuario puede escribir una sentencia en términos del Álgebra Relacional de una manera muy cómoda. Estas operaciones son: Selección, Proyección, Intersección, Unión, Cociente, Reunión (Join), Producto Cartesiano, Diferencia y Renombrado.

En la Figura 2 aparece esta ventana con las operaciones disponibles.

En la aplicación también se cuenta con otra ventana en la que aparecen una serie de pestañas que muestran información de cierto interés para el usuario. Esta ventana aparece en la Figura 3. Las

pestañas son las siguientes:

- Errores*: Muestra la descripción de los errores que se hayan podido producir durante el proceso de compilación.
- SQL*: Muestra las instrucciones SQL resultantes del proceso de compilación si éste ha terminado con éxito.
- Tablas*: Muestra la descripción de la base de datos con los nombres de las tablas y los atributos correspondientes.

Por último, la ventana de edición permite al usuario introducir la sentencia del Álgebra Relacional, visualizando cada operador con la fuente adecuada.

En la Figura 4 aparece la ventana de edición con una consulta expresada en términos del Álgebra Relacional y que permite proporcionar el nombre, apellidos y dirección de todos los empleados que trabajan en el Departamento que tiene por nombre "Investigación".

Los resultados de la consulta realizada por el usuario aparecen en otra ventana. Allí aparecen las tuplas que pertenecen a la relación definida por la expresión introducida por él.

Los resultados, evidentemente, no pueden ser modificados, pero a la vista de lo obtenido, el alumno puede modificar su consulta, o introducir nuevos datos en las tablas de la base de datos que le permitan aceptar o rechazar la expresión que haya construido como respuesta a la necesidad de información previamente planteada por él o por una serie de cuestiones que le son planteadas en las sesiones de laboratorio.



Figura 3. Ventana de información



Figura 4. Ventana de edición

## 5. Trabajo futuro

Como ya hemos comentado, nos encontramos ante la primera versión de esta herramienta. Es necesario recordar que nos planteamos el diseño de este tipo de software como un proceso evolutivo y dinámico, de manera que una de nuestra líneas de trabajo se centra en medir la efectividad de dicha herramienta en el proceso de aprendizaje de los alumnos.

El diseño del experimento para llevar a cabo esta tarea nos hace pensar en la comparación de dos tratamientos: aprendizaje usando la herramienta y aprendizaje sin usarla (como se estaba haciendo antes de incorporarla). Esta tarea se desarrollará durante el primer cuatrimestre del curso 2002/2003.

Otra línea de trabajo se centra en revisar el diseño inicialmente realizado para intentar eliminar la necesidad de utilizar un determinado SGBDR, como ocurre en la versión actual, incorporando tecnología *JDBC/ODBC*, lo que indudablemente incrementaría la disponibilidad de la herramienta. Este trabajo se ha acometido mediante la propuesta de un Proyecto Fin de Carrera, que ya se está realizando en la actualidad y que previsiblemente estará terminado en el mes de Julio de este año.

Por último, mantenemos una línea de trabajo abierta para posibilitar el uso de la herramienta en la World Wide Web. De esta manera, enlazaríamos este trabajo con otro que ya está en marcha y que pretende la construcción de un portal web en castellano para el aprendizaje de

lenguajes de manejo de bases de datos. Para la realización de este proyecto se ha solicitado una subvención a la Junta de Castilla y León dentro de su “Programa de Ayudas para la Elaboración de Recursos de Apoyo y Experiencias Innovadores en la Enseñanza Universitaria”.

## 6. Conclusiones

En este trabajo se ha presentado una herramienta que facilita el aprendizaje del Álgebra Relacional. Se trata de un sistema capaz de realizar consultas sobre cualquier base de datos. Ante una sentencia expresada en Álgebra, el sistema traduce esta sentencia a SQL. Esta traducción se muestra al alumno haciéndole más patente la relación entre ambos lenguajes. Se le muestran los operadores que puede utilizar y las tablas, junto con sus atributos, que puede utilizar para hacer la consulta. Además, se permite una realimentación inmediata, al incluir la posibilidad de reescribir su consulta, de manera que puede ver el resultado de la consulta que había propuesto y puede realizar las oportunas modificaciones.

La herramienta ha sido desarrollada siguiendo guías metodológicas propugnadas en el ámbito del diseño de entornos de aprendizaje, para garantizar que cumple los fines con los que ha sido concebida.

## Referencias

- [1] Boehm, B., "A Spiral Model for Software Development and Enhancement". *Computer*, vol. 21, n° 5, pp. 61-72.
- [2] Dietrich, S.W., Eckert, E., Piscator, K., "WinRDBI: A Windows-based Relational Database Educational Tool", *Proceedings of the 28<sup>th</sup> ACM SIGCSE Technical Symposium on Computer Science Education*, San Jose, California, February 27-March 1, 1997, pp. 126-130.
- [3] Dietrich, S.W., Urban, S.D., "Cooperative Learning Approach to Database Group Projects: Integrating Theory and Practice", *IEEE Transactions on Education*, November 1998, CD-ROM Directory 06, 11p.
- [4] Furlán, A., "Metodología de la enseñanza". *Aportaciones a la didáctica en la Educación Superior*, UNAM-ENEP, Iztacala, México, D.F. 1989.
- [5] Gewerc, A., "Diseño de entornos de aprendizaje", <http://www.quadernsdigitals.net/articles/quadernsdigitals/quaderns24/q24diseno.htm>. 2002
- [6] Gros, B., "*Diseño de programas educativos*", Barcelona, Ariel, 1997.
- [7] Quintana, R.A., Valentín, T., "*Diseño e implementación de un Intérprete del Álgebra Relacional para uso docente*", P.F.C., Universidad de Valladolid, Enero 2002.