

Base de Datos y Teoría de Modelos Finitos
VIII Ateneo de Profesores Universitarios de Computación

Gagliardi, Edilma Olinda
Herrera, Norma Edith
Piccoli, María Fabiana
Reyes, Nora Susana
{ oli, nherrera, mpiccoli, nreyes } @unsl.edu.ar

Asesores

Ing. Hugo Ryckeboer
Dr. José María Turull Torres

Universidad Nacional de San Luis
Departamento de Informática
Área de Datos
Ejército de Los Andes 950
San Luis, Argentina

Resumen

La enseñanza de la disciplina Base de Datos tiene como objetivos que el licenciado pueda desarrollarse sólidamente en el campo profesional, adaptándose con facilidad a los rápidos cambios tecnológicos propios de la materia, como también iniciarse en la investigación científica y desarrollos propios de su área de interés.

En el Plan de Estudios de la Licenciatura se ha considerado el estudio del tema comenzando desde los inicios de la carrera hasta su finalización, de forma tal que la formación sea paulatina, relacionada con los demás tópicos de las asignaturas correspondientes y finalmente, en el último curso, introduciendo un marco teórico profundo y adecuado que abarque en la forma más completa posible las aplicaciones reales y que admita la consideración de problemas en la computación de consultas o queries.

Equipo de Cátedra: Gómez, Alejandro Javier; Ludueña, Verónica del Rosario;
Taranilla, María Teresa.

Introducción

La carrera Licenciatura en Ciencias de la Computación de la Universidad Nacional de San Luis persigue como objetivos que el licenciado pueda dedicarse tanto a la actividad profesional como a la científica, con herramientas teóricas, metodologías adecuadas y aptitudes críticas que le permitan un apropiado y competente desempeño en su actuación de acuerdo a los requerimientos eventuales.

En este sentido, con respecto a la disciplina Base de Datos, se busca cubrir un núcleo básico en los aspectos teóricos y prácticos lo suficientemente amplio. De esta manera, el alumno recibe una visión comprensiva de la materia, sustentada en una formación teórica que le permite una constante actualización mediante el uso de literatura científica actual y una capacidad de adaptación a los rápidos cambios tecnológicos.

La enseñanza de la disciplina está distribuida en tres asignaturas dictadas en el Área de Datos del Departamento de Informática, las cuales están relacionadas en su temática y son correlativas en el plan de estudio.

Desde el año 1999 está vigente el nuevo plan de la carrera, en el cual estas tres asignaturas sufrieron cambios de contenidos buscando adaptarse a la realidad contemporánea, dados los cambios existentes en la última década en lo que a computación se refiere. En el tópico Bases de Datos, prácticamente se han respetado los contenidos del plan anterior, agregándole la vinculación más estrecha a Teoría de Modelos Finitos. También, la incorporación de nuevas asignaturas al Plan nuevo tales como Lógica, Complejidad y Computabilidad, permiten establecer una conexión más clara con la Teoría de Base de Datos.

En el siguiente cuadro mostramos cómo se corresponden las materias entre el nuevo y viejo plan y cuáles son los respectivos años de dictado.

Nivel	Plan Anterior	Plan Nuevo
1	Estructuras de la Información (2° año)	Estructuras de Datos y Algoritmos (2° año)
2	Organización de Archivos (3° año)	Organización de Archivos y Bases de Datos I (3° año)
3	Recuperación de la Información (5° año)	Base de Datos II (4° año)

El hecho de nombrar niveles sirve a los efectos de mostrar las instancias por las que el alumno va pasando y encontrándose con contenidos de Base de Datos.

Las tres asignaturas presentadas, en cualquiera de los planes, son correlativas dada la temática que tratan. Los contenidos que se incorporan en cada una respecto de la disciplina Base de Datos son relacionados y progresivos en su profundización y complejidad.

Enfoque elegido

Se pretende introducir la disciplina por etapas, específicamente tres niveles, en donde cada uno revisa en mayor profundidad los contenidos vistos en el nivel anterior, aumentando la complejidad de los temas y agregando nuevos tópicos relacionados. De esta manera, el conocimiento se adquiere en forma paulatina, con una sólida base y se alcanza un estado de comprensión íntimamente relacionado a la evolución del aprendizaje del alumno.

En el primer nivel se introducen conceptos básicos de la Teoría de Base de Datos, buscando que el alumno intuitivamente comience a usar herramientas formales que se requieren para el diseño, desarrollo e implementación de sistemas.

Ya en el segundo nivel, la teoría básica de Modelo Relacional es cubierta, con bibliografía clásica [AHV95][Mai83] [Ull82] [Ull88] y se presenta un lenguaje formal de consultas a bases de datos relacionales: Álgebra Relacional. Estos tópicos se completan vinculándolos con productos de software comerciales para el manejo de Bases de Datos Relacionales.

Así, en este punto, si el alumno opta por obtener solamente el título intermedio, tiene la idoneidad suficiente en la disciplina, contando con los conceptos, principios y teorías que constituyen el ámbito de competencia. Con ello podrá brindar soluciones a problemas de diversas aplicaciones, eventualmente continuar su propio perfeccionamiento en el ámbito profesional y desarrollarse como analista-programador mediante la experiencia práctica.

En el caso que el alumno persiga obtener el grado de licenciado, entonces se prevé una orientación teórica mucho más fuerte, siguiendo la misma metodología de retomar lo visto y revisado en niveles anteriores. Se busca además agregar herramientas formales, formalismos más estrictos y estudiar problemas usuales en el ámbito de la informática, que no son sustentados por una teoría de base sólida. El hecho de mostrar estos desarrollos teóricos apunta a evitar los posibles conflictos que se desprenden de la inadecuación de los motores de bases de datos existentes, brindando la posibilidad de definir pautas de diseño en las bases de datos y en las consultas.

En este nivel mediante la Teoría de Grafos, la Lógica y la Teoría de Modelos Finitos, se muestran los diferentes modelos de bases de datos, paradigmas de lenguajes formales de consultas y los problemas de expresividad relacionados a diversos lenguajes.

Alcance de los contenidos

A continuación hacemos una descripción de la distribución de los contenidos que se imparten respecto de la disciplina, mostrando los objetivos buscados en cada nivel y la actuación del alumno frente al estudio e investigación del área científica y práctica.

Nivel 1

En este nivel se ataca el problema del diseño de Estructuras de Datos desde un enfoque relacional.

Para ello se analiza cómo el problema de acceder a la información puede verse como el problema de acceder a nuplas de una relación $R \subseteq X \times Y$, siendo X e Y dominios conceptuales.

En particular, se estudia el caso de consultas o evocaciones asociativas: conocemos una parte de la nupla, X, y queremos recuperar el resto, Y. En este punto, se introduce el concepto de dependencia funcional, dado que influye en el diseño de las estructuras, en las rutinas que resuelven la evocación y en la interfaz con el programa. Se pueden analizar dos tipos de evocaciones asociativas:

- Evocación de respuesta única, ocurre cuando X determina unívocamente a Y
- Evocación de respuesta múltiple, ocurre cuando X no determina a Y

Desde este punto de vista se introducen conceptos básicos de Bases de Datos Relacionales, formas normales y su justificación, haciendo ver la necesidad y las ventajas de mantener las relaciones normalizadas, antes de pensar en las estructuras de datos necesarias para resolver las consultas o evocaciones planteadas sobre las relaciones. Como método de normalización, sólo se estudia el método de Descomposición. También se analizan algunos de los operadores primitivos del Álgebra Relacional, como la proyección y el ensamble de relaciones (join).

El alumno adquiere así los primeros conceptos e ideas vinculadas al Modelo Relacional.

Nivel 2

Considerando como punto de partida lo visto en el nivel 1, se retoman las dependencias funcionales observándolas como restricciones de una realidad y se introduce el diseño formal de Bases de Datos Relacionales [Cod70] [Mai83] [Ull82] [Ull88], considerando diversos métodos de diseño que aprovechan la presencia de dependencias y consideran la preservación de propiedades deseables en un buen diseño. También se realiza la presentación de un lenguaje formal, Álgebra Relacional, para la expresión de las consultas a bases de datos relacionales.

Las dependencias se presentan como un sistema formal, en el que se consideran los aspectos de interpretación y demostración, introduciendo la relación con Lógica.

Para el diseño de bases de datos relacionales, se estudian dos métodos: Síntesis y Descomposición, mostrando las virtudes y posibles anomalías que los mismos pueden guardar frente al análisis de la realidad. Respecto de las propiedades, se estudian las formas normales, la no pérdida de información y la preservación de dependencias, como así también diversos métodos de verificación para determinar la satisfacción de las mismas.

Referido a la expresión de las consultas, se introduce el Álgebra Relacional, considerando en su estudio las características propias del lenguaje y las posibles estructuras de las consultas. Además, se introduce el concepto de expresividad de los lenguajes. En un análisis más fino y agudo, mostramos clases de consultas no expresables por el Álgebra Relacional, tales como las consultas relacionadas a clausura transitiva, cardinalidad, funciones aritméticas, etc.

Con esta base teórica, se presentan ejemplos de administradores de bases de datos relacionales mostrando de qué manera se define en ellos una base de datos y cómo se pueden realizar las consultas en un lenguaje de aplicación y comercial, presentando para ello el lenguaje SQL y estableciendo una analogía con Álgebra Relacional.

Los alumnos, por su parte, forman grupos de trabajo con el fin de realizar trabajos prácticos y de investigación. Una tarea consiste en presentar informes sobre los temas de estudio, introduciéndose así en más detalles que los que la bibliografía habitual presenta. Por ejemplo, en el

caso del Álgebra Relacional, un grupo toma un operador, lo define, analiza sus casos extremales (instancias y/o esquemas vacíos, etc.) y presenta ejemplos de consultas de mayor complejidad que las habitualmente vistas. También describe un algoritmo utilizando los operadores abstractos necesarios para el manejo de las estructuras de datos¹ y analiza la complejidad computacional que posee el operador expresándola en notación O (Big-Oh notation). De la misma manera, analizan las limitaciones de expresividad del Álgebra Relacional.

Finalmente un trabajo integral es requerido, el cual consiste en la presentación de una realidad, el diseño de la base de datos relacional y la metodología aplicada, junto con las demostraciones correspondientes. También presentan posibles consultas o servicios requeridos y sus respectivas resoluciones en Álgebra Relacional.

Con esto se considera que si el alumno opta por terminar su carrera en este punto, ha obtenido las herramientas teóricas y el acercamiento necesario a administradores de bases de datos relacionales para su desempeño laboral.

Nivel 3

Continuando con la metodología de estudio presentada, en este nivel se revisan los temas vistos en los niveles anteriores y se realiza un ajuste sobre los mismos, profundizándose en más detalles o extendiéndolos. Se agrega el estudio de otros modelos o paradigmas de diseño de bases de datos, tales como los modelos de Red, Jerárquico, Valores Complejos, etc. y lenguajes formales de consultas a bases de datos. Asimismo, se establece una vinculación más estrecha de la Teoría de Base de Datos con otras disciplinas, que le sirven de marco teórico formal.

Comenzamos retomando el estudio del Modelo Relacional, e incorporando otros tipos de dependencias tales como multivaluadas, incluidas y otras, y sus respectivas soluciones al diseño.

Se hace una apertura al estudio de otros modelos de bases de datos, tales como Modelo Algebraico (formalización del modelo Entidad-Relación), Red, Jerárquico, etc. Se presenta para cada modelo un administrador de base de datos conveniente (DBTG y DMSII o IMS). Se estudian las estructuras de datos subyacentes en cada caso y se hace la relación con los demás tópicos vistos en los niveles anteriores. Nuevamente, haciendo uso de los operadores abstractos para el manejo de estructuras de datos, se presentan algoritmos para actualización y consultas a las bases, según diversos casos de estudio, mostrando la complejidad correspondiente. Para una mejor integración de lo visto, se realiza una comparación analítica entre los modelos estudiados, mediante la Teoría de Grafos.

A partir de aquí y continuando con el Modelo Relacional, se estudian las consultas, *queries*, a bases de datos, usando como marco teórico la Teoría de Modelos Finitos [AHV95] [Cai90] [Ebi85] [EF95] [Flu85] [Gur84].

En Teoría de Modelos Finitos, el Modelo Relacional se formaliza observando el esquema de la base de datos relacional como un vocabulario relacional finito σ , donde cada símbolo de relación en σ es de una cierta aridad; y la instancia de base de datos relacional es una σ -estructura, donde cada relación es de la aridad correspondiente para cada símbolo de σ , definida en un dominio finito dado. De esta manera, se consideran a las bases de datos relacionales como clases de estructuras

¹ Herramientas obtenidas en el nivel 1 al estudiar las evocaciones asociativas

relacionales finitas, y a las instancias de bases de datos relacionales como estructuras relacionales finitas.

En [CH80] se propone como definición de la clase de queries computables a la clase de funciones definidas en las clases de estructuras relacionales finitas, para cada vocabulario relacional finito, tales que son funciones recursivas parciales en alguna codificación lineal de las estructuras, y preservan isomorfismos en ellas. A dicha clase se la llamó **CQ**, por *Computable Queries*. Desde esta visión, un query se ve como una función cuyo dominio es el conjunto de las estructuras relacionales finitas de un cierto vocabulario, y cuyo codominio es el conjunto de las relaciones definidas en el dominio de la estructura relacional finita correspondiente para alguna aridad, $f: \mathcal{E}_{\sigma, Fin} \rightarrow \mathcal{E}_{\langle R \rangle}$.

Si σ es un vocabulario relacional finito, $\varphi \in L_{\sigma}$ ² con variables libres $\{x_1, x_2, \dots, x_r\}$, y B es una σ -estructura, con $d_1, \dots, d_n \in D^B$, denotaremos con la siguiente expresión (\models) al valor de verdad Verdadero de la fórmula φ interpretada en la estructura B, con el elemento d_{s_j} asignado a la variable libre x_j , para $1 \leq s_j \leq n$, $1 \leq j \leq r$, entonces $B \models \varphi(x_1, \dots, x_r)[d_{s_1}, \dots, d_{s_r}]$. La acción de asignar el elemento d_{s_j} a la variable libre x_j , siendo x_j variable libre, es denominada valoración.

Nótese que de acuerdo a la semántica del lenguaje considerado, φ define o expresa un query r-ario f_{φ} sobre la estructura B. Es decir, $f_{\varphi}(B)$, denotado por φ^B , es una relación finita definida en la estructura B por la fórmula φ , cuya aridad estará determinada por la cantidad de variables libres de φ . En símbolos:

$$\varphi^B = \{ (d_{s_1}, \dots, d_{s_r}) : d_{s_1}, \dots, d_{s_r} \in D^B \wedge B \models \varphi(x_1, \dots, x_r)[d_{s_1}, \dots, d_{s_r}], 1 \leq s_j \leq n, 1 \leq j \leq r \}$$

Hasta el momento se han utilizado principalmente cuatro tipos de formalismos para el estudio de los queries computables: lenguajes de programación formales (QL [CH80]), máquinas abstractas (GMloose y GM [AV91]), lógicas abstractas [EBI85] [EFT84] [Gur84] [Imm87] y circuitos booleanos [BDG88] [BDG90] [Voll99]. Nosotros nos dedicamos en este nivel a Lógica de Primer Orden, para abordar el estudio del Cálculo Relacional y sus problemas de expresiones seguras; a un lenguaje de programación formal completo: QL, y se presentan los trabajos de investigación relacionados a los cuatro tipos de formalismos mencionados anteriormente, realizados por docentes del área.

Como problemática de la computación de queries o consultas a bases de datos muy grandes o distribuidas, mostramos que se requiere de un marco teórico sólido que provea herramientas formales para su consideración. Véanse al respecto los problemas descritos en [Tur95] con relación a la inadecuación de la Teoría de Computabilidad clásica con respecto a la computación de queries a bases de datos. Esto provoca problemas prácticos porque los motores de bases de datos permiten expresar queries que no preservan isomorfismos (como obtener los datos de "un" cliente que cumpla cierta propiedad). Si se desea parcializar la evaluación de tal query (por ejemplo, porque su evaluación requiere mucho tiempo de procesador) y realizar una reorganización de la base de datos entre dos computaciones parciales, se pueden lograr resultados inconsistentes, y esto es porque dichos motores no respetan el concepto de query computable [GGHR 00].

Como propuesta adicional, se agrega el estudio de los Valores Complejos [AHV95], como un posible marco teórico para el estudio de las Bases de Datos Orientadas a Objetos y de las Bases de

² L_{σ} es el lenguaje de primer orden con vocabulario σ

Datos de Valores Complejos, como así también el estudio de la computación de consultas a Bases de Datos de Valores Complejos. El modelo de Valores Complejos ha sido propuesto como una extensión al Modelo Relacional, para dar solución a los problemas presentados por éste último. Los Valores Complejos son relaciones en las cuales las componentes de las tuplas no son necesariamente atómicas, sino que en sí pueden ser relaciones y/o conjuntos. También el Modelo de los Valores Complejos proporciona un marco formal adecuado para las Bases de Datos Jerárquicas y de Red, puesto que para realidades cuyos datos están estructurados jerárquicamente, resulta natural modelarlas con Valores Complejos.

Finalmente, para conectar teoría y práctica, los alumnos conforman grupos de trabajo en donde eligen un software administrador de bases de datos relacionales, toman una realidad, diseñan la base de datos relacional y plantean posibles servicios o requerimientos de la misma, resolviéndolos en el SQL del software elegido. Como los alumnos ya poseen conocimientos de Sistemas Operativos, fácilmente pueden relacionar y comprender el funcionamiento de un administrador de bases de datos relacionales en máquinas de mayor envergadura que un simple Pc. Dicho trabajo de investigación lo efectúan por su cuenta, tomando ellos mismos conciencia de que, aún faltos de experiencia, están capacitados para abordar el estudio y aprendizaje de un software particular y conseguir la resolución de problemas de bases de datos con mayor precisión o abstracción, según sea el caso.

Actividades relacionadas al estudio de la disciplina

Para alimentar y consolidar esta línea de estudio, el Área de Datos proporciona diferentes actividades integradoras en el estudio de la disciplina.

Se cuenta con un Seminario permanente de Teoría de la Computación, en donde los docentes tienen un ámbito de estudio y discusión sobre los avances teóricos de la disciplina.

Como carrera de postgrado existe la Maestría en Ciencias de la Computación con orientación en Teoría de la Computación, dirigida por el asesor Dr. Turull Torres.

También, en el marco del Proyecto FOMEC, han sido invitados profesores de otras Universidades, tales como Dr. Lauri Hella (Universidad de Helsinki, Finlandia), Dr. Michal Krynicki (Universidad de Varsovia, Polonia) y Dr. Xavier Caicedo (Universidad de Los Andes, Colombia), entre otros, quienes son exponentes renombrados en la temática Teoría de Modelos, en particular Teoría de Modelos Finitos, para dictar cursos de postgrado, manifestar el estado del arte de la disciplina en la actualidad y participar en tareas de investigación conjuntas.

Los docentes del área dirigen trabajos de fin de carrera de la licenciatura, consistentes en la programación de algoritmos relacionados a algún tópico formal particular; por ejemplo, como el caso de la implementación de un interprete para el lenguaje formal QL.

Actualmente, se está evaluando la posibilidad de incorporar Talleres de Bases de Datos, extracurriculares, para ofrecer un ámbito al estudiante que haya cursado al menos el 2° nivel, donde pueda experimentar la utilización de un sistema administrador de bases de datos de alta complejidad.

Conclusiones

El punto esencial es que en la medida en que la Informática pretenda atacar problemas de alta complejidad, lo que se ha tornado habitual debido al tremendo desarrollo en cuanto a tecnología informática, se hace más necesaria una teoría de base, sólida y dinámica, que permita utilizar herramientas formales para enfrentar esos problemas. De allí que pensemos que es altamente relevante hoy en día el desarrollo de las teorías de base que permitan comprender cabalmente los fenómenos que subyacen en las distintas áreas de las Ciencias de la Computación.

Por otra parte, estos desarrollos teóricos, nos permiten definir pautas de diseño en las bases de datos que apuntan a evitar los posibles conflictos que se desprenden de la inadecuación de los motores de bases de datos existentes a la teoría antes mencionada.

Por lo tanto, creemos que es adecuado el enfoque elegido en la enseñanza de la disciplina, puesto que aunque el avance tecnológico sea rápido y el avance teórico más lento, permite que el licenciado ataque los problemas que subyacen permanentemente, con una base teórica sólida y confiable.

Referencias Bibliográficas

- [AHV95] Abiteboul,S; Hull and Vianu, V.; “Foundations of Databases”. Addison-Wesley Publishing Company, 1995.
- [AV91] Abiteboul,S; Vianu, V.; “Generic Computation and Its Complexity”, STOC 1991.
- [BDG88] Balcázar, J.L.;Díaz, J; Gabarró, J; “Structural Complexity I”. Springer- Verlag, 1988.
- [BDG90] Balcázar, J.L.;Díaz, J; Gabarró, J; “Structural Complexity II”. Springer- Verlag, 1990.
- [Cai90] Caicedo, X.; “Elementos de Lógica y Calculabilidad”, Empresa Docente, Universidad de los Andes, Colombia.
- [CH80] Chandra, A.K.; Harel, D. “Computable Queries for Relational Data Bases”. Journal of Computer and System Sciences 21, 156-178. 1980.
- [Cod70] Codd, E.F.; “A relational model of data for a large shared data banks”. Com of ACM 13(6):377-387,1970.
- [Ebi85] Ebbinghaus, H.; “Extended Logics: The General Framework”, in Perspectives in Mathematical Logic, Springer-Verlag, 1985.
- [EF95] Ebbinghaus, H; Flum, J.; “Finite Model Theory” , Springer-Verlag, 1995.
- [EFT84] Ebbinghaus, H; Flum, J.;Thomas, W.; “Mathematical Logic”, Springer-Verlag, 1984.

- [Flu85] Flum, J.; “Characterizing Logics”, in Perspectives in Mathematical Logic, Springer-Verlag, 1985.
- [GGHR00] Gagliardi, E; Grosso, A; Herrera, N; Reyes, N. “Teoría de la Computación y Bases de Datos”, WICC 2000.
- [Gur84] Gurevich, Y.; “Toward logic tailored for computational complexity”, in Computation and Proof Theory, Lecture Notes in Mathematics 1104, 1984.
- [Imm87] Immerman, N.; “Languages that Capture Complexity Classes”, SIAM J. on Computing 16, 4; 1987.
- [Mai83] Maier, "The theory of relational databases", Maier. Computer science press, 1983.
- [Tur95] Turull Torres, J.M.; “De la Computabilidad sobre Números Naturales a la Computabilidad sobre Estructuras o Bases de Datos”, INFOCOM 95, Junio 1995.
- [Ull82] Ullman, Jeffrey D. "Principles of database systems", vol 1, Computer Science Press, 1982.
- [Ull88] Ullman, Jeffrey D. “Principles of Database and Knowledge Base Systems”. Computers Science Press, 1988.
- [Voll99] Vollmer, Heribert, “Introduction to Circuit Complexity, a uniform approach”. Springer Verlag, 1999.