



DBDAp: Una herramienta de apoyo a la docencia de bases de datos y almacenes de datos

Sergio Ilarri

Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura
I3A, Universidad de Zaragoza
silarri@unizar.es

Resumen

En este artículo se presenta DBDAp, una herramienta gráfica de apoyo a la docencia de bases de datos y almacenes de datos. La herramienta permite definir esquemas conceptuales de bases de datos y transformarlos automáticamente a esquemas relacionales apropiados. La herramienta desarrollada tiene principalmente una finalidad educativa e incluye en consecuencia un modo de transformación paso a paso junto con la posibilidad de acceder a explicaciones, consejos y sugerencias, así como capacidad para la detección de posibles problemas de modelado. Incorpora también soporte para la definición de esquemas en estrella para almacenes de datos y diferentes notaciones gráficas alternativas para el diseño conceptual de bases de datos, tales como el modelo E/R y el lenguaje UML.

Palabras clave: Bases de datos, almacenes de datos, diseño de bases de datos, modelo E/R, modelo relacional, esquemas en estrella.

1. Introducción

Las bases de datos son un elemento central de los sistemas de información. Esto no sólo es una realidad en la industria, sino que es enfatizado por una diversidad de trabajos: «*Databases, the center of today's information systems*» [9]; «*Most modern information systems utilize database technologies*» [24]; «*The database is now the underlying framework of the information system*» [15]. Por tanto, es importante formar adecuadamente a los estudiantes en todas aquellas habilidades relacionadas con la definición, administración y explotación de bases de datos.

En el contexto de asignaturas relacionadas con la temática de bases de datos, puede observarse que el diseño de bases de datos [17, 20] es una de las tareas que supone mayores dificultades para el alumnado. En particular, la fase de diseño conceptual (ilustrada en la figura 1) requiere una capacidad de abstracción que el alumnado debe ir desarrollando poco a poco. A los retos de conceptualización se añaden confusiones referentes a la interpretación o utilización de la notación, así como en lo concerniente a las reglas de transformación

de los esquemas conceptuales en esquemas relacionales (fase de diseño lógico). Trabajos como el de Connolly y Begg [15] señalan la importancia de seguir una *aproximación constructivista (aprendizaje basado en problemas y aprendizaje basado en proyectos)* para el aprendizaje del análisis y diseño de bases de datos, pero en general no existen buenas herramientas de soporte para la realización de los diseños. Por ello, decidimos iniciar el desarrollo de DBDAp, una herramienta gráfica de apoyo a la docencia de bases de datos y almacenes de datos que, además de tener una orientación docente de apoyo al profesorado, estuviera también enfocada al alumnado como ayuda para el aprendizaje de aquellos aspectos que habitualmente resultan complicados.

La herramienta sirve de apoyo al diseño de bases de datos, pero sobre todo pretende ser una herramienta para ayudar en el aprendizaje y desarrollo de las habilidades requeridas para el diseño de bases de datos. Así, permite definir esquemas conceptuales utilizando el modelo Entidad/Relación (E/R) [11] o el lenguaje de modelado UML [43], transformarlos a esquemas relacionales, normalizar los esquemas siguiendo la teoría de la normalización de bases de datos relacionales [26, 46],

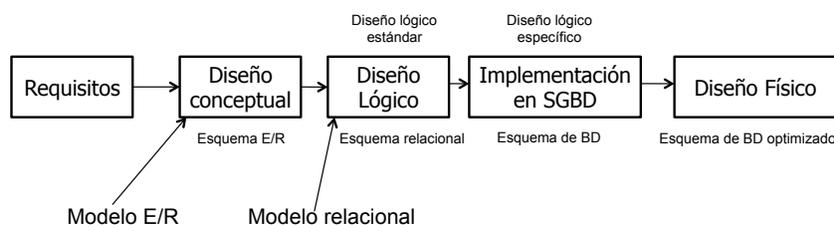


Figura 1: Esquema general de la metodología de diseño de una base de datos

etc. Dado que la herramienta tiene principalmente una finalidad educativa, incluye un modo de transformación paso a paso de esquemas E/R en esquemas relacionales, junto con la posibilidad de acceder a explicaciones, consejos y sugerencias, así como soporte para la detección de posibles problemas de modelado. Además, la herramienta también incluye soporte para la definición de esquemas en estrella para el diseño de almacenes de datos.

Aunque existen diversas aplicaciones informáticas orientadas al diseño y desarrollo de bases de datos (CA ERwin Data Modeler, TabletERD, GNU Ferret, ERMaster), no se han encontrado herramientas gratuitas que ofrezcan un rango de funcionalidades tan amplio como el disponible en la herramienta desarrollada. Además, no se conoce ninguna que esté específicamente pensada para enseñar esta materia. Como se expondrá a lo largo del artículo, la herramienta aglutina una serie de funcionalidades que resultan de gran interés para una variedad de asignaturas de la temática de bases de datos.

2. Aspectos básicos de diseño

Se pueden identificar diversos elementos, relacionados con el diseño de bases de datos y almacenes de datos, para cuyo aprendizaje resulta de gran utilidad disponer de una herramienta como la presentada en este artículo. En el resto de esta sección analizamos brevemente dichos aspectos.

2.1. Diseño conceptual de bases de datos

Como modelo de datos para el diseño conceptual de bases de datos se utiliza habitualmente el modelo *Entidad/Relación* [11] (*E/R*, también conocido como *Entidad/Interrelación*, *Entidad-Relación*, o *E-R*). Además, se incluyen las diversas extensiones que se incorporaron posteriormente sobre el modelo básico, dando lugar a lo que habitualmente se conoce como *modelo E/R extendido* [20]. Las extensiones propuestas sobre el modelo básico suponen la introducción de conceptos de modelado semántico adicionales como las especializaciones/generalizaciones (herencia). Normalmente, como haremos en este artículo, se habla en general del modelo E/R dando a entender que se trata del modelo básico junto con las extensiones al mismo habitualmente consideradas.

Una alternativa popular es utilizar diagramas de clases en *UML* [43] (*Unified Modeling Language*). No obstante, al menos para cursos introductorios en bases de datos, se puede decir que hay un mayor soporte en favor de la utilización del modelo E/R para el diseño conceptual [7, 42, 44, 49].

Así, habitualmente se prefiere utilizar el modelo E/R en lugar de UML. La principal razón es que el modelo E/R es el método más popular para el diseño conceptual de bases de datos mientras que UML se usa sobre todo en el análisis y diseño orientado a objetos. Eso hace que, por ejemplo, sea necesario incorporar notaciones específicas en forma de *estereotipos* a los diagramas de clase UML utilizados para modelar bases de datos (por ejemplo, para representar claves primarias, claves ajenas, etc.). En general, se considera que el modelo E/R extendido proporciona una mayor potencia expresiva para el diseño conceptual de bases de datos.

No obstante, sería deseable poder disponer de una herramienta que permita realizar el diseño conceptual utilizando ambas aproximaciones (E/R y UML) e incluso que permita pasar automáticamente de una a otra representación. La motivación es doble. Por un lado, de este modo, el alumnado podrá percibir mejor la relación existente entre ambas aproximaciones. Por otro lado, para algunas asignaturas resulta relevante considerar UML como herramienta de trabajo adicional, especialmente si en dichas asignaturas se introducen también las características de orientación a objetos (bases de datos orientadas a objetos y objeto-relacionales).

Sería también deseable que la herramienta informática que proporcionara el soporte para realizar los diseños de bases de datos ofreciera también una funcionalidad para realizar transformaciones automáticas entre ambas representaciones (E/R y UML). Esto ayudaría al alumnado a entender claramente las relaciones entre ambos tipos de modelos, evitando potenciales confusiones. Así, de acuerdo con Neubauer [39], «*if the presentation of multiple modeling techniques is not carefully planned, students may confuse terms and mix modeling notations*».

Además, disponer de una herramienta que soporta diferentes notaciones permite disminuir el impacto de la elección de una u otra representación. En trabajos tales como el de Martínez-González y Duffing [33] se enfatiza que la verdadera dificultad para los estudiantes no está en el modelo o notación utilizada, sino en la dificultad de conceptualización que supone el diseño conceptual. No hay que olvidar que el diseño

de bases de datos es una tarea compleja, dado que para su correcto desarrollo es preciso progresar hasta al menos el cuarto nivel de la *taxonomía de Bloom* [18, 27] (*análisis*) e incluso hasta el quinto nivel (*síntesis*), superando los niveles iniciales de *conocimiento*, *comprensión* y *aplicación*; si esto lo combinamos con actividades de evaluación entre pares o con tareas de análisis crítico de diseños ya realizados, podríamos incidir también en el sexto nivel de *evaluación*. Por tanto, lo más importante es ayudar a los estudiantes a que desarrollen esas destrezas de abstracción y conceptualización necesarias y que puedan utilizar, si así lo requieren dichos estudiantes o las características de la materia tratada, diversas notaciones.

2.2. Diseño lógico de bases de datos

El *modelo relacional*, presentado por E.F. Codd en 1970 [14], se considera habitualmente el modelo de datos para el diseño lógico dada su utilización masiva y su clara supremacía sobre otros modelos de datos (jerárquico, en red) [2]. No obstante, en algunas asignaturas también se considera un modelo relacional con características de orientación a objetos: el modelo objeto-relacional [19].

Las reglas de transformación que deben aplicarse para pasar de un esquema conceptual utilizando el modelo E/R extendido a un esquema lógico utilizando el modelo relacional son bien conocidas y pueden automatizarse en buena medida (por ejemplo, ver [51]); una excepción notable es que existen diversas maneras de traducir una especialización/generalización (relación de herencia o jerarquía) [3, 20], como se comenta en el apartado 3.3.1. Sin embargo, la aplicación de dichas reglas causa dificultades al alumnado, especialmente cuando el esquema E/R incluye elementos tales como tipos de interrelación ternarias o tipos de entidades débiles. También es habitual que el alumnado olvide añadir restricciones en lenguaje natural que garanticen propiedades necesarias de los datos que en ocasiones se pierden al pasar de utilizar un modelo semánticamente rico como es el modelo E/R a un modelo relacional plano.

2.3. Diseño lógico de almacenes de datos

Para el diseño lógico de almacenes de datos, se considera el tradicional *modelo en estrella* [1, 28, 30], que sigue las reglas del modelo relacional pero con algunas guías de diseño diferentes con objeto de adecuar la estructura del repositorio de datos a tareas de análisis en lugar de al procesamiento transaccional. Los dos objetivos básicos que rigen dichas reglas de diseño son:

- **Simplicidad:** esquemas fácilmente interpretables y utilizables por usuarios no expertos. Dichos esquemas deberían ser fácilmente accesibles incluso para usuarios no técnicos de la empresa donde se ha implantado. Por eso se busca construir esquemas sencillos y simétricos.
- **Eficiencia:** rapidez en la ejecución de consultas analíticas sobre grandes volúmenes de datos históricos.

Este doble objetivo, junto con el propósito particular y contexto de utilización de los almacenes de datos, hace que, por ejemplo, algunas estrategias de diseño de almacenes de datos se opongan frontalmente a lo que se haría en un diseño relacional de una base de datos tradicional. Por citar dos ejemplos, de acuerdo con la metodología de diseño de Ralph Kimball, los esquemas de un almacén de datos suelen estar desnormalizados, para evitar anomalías de inserción, borrado y actualización gracias a la implementación del proceso de carga a través de un único proceso ETL (*Extraction, Transformation, Load*) [29], bien diseñado para garantizar la integridad de los datos. También se utilizan claves artificiales: claves numéricas que no tienen sentido en el dominio del problema, con objeto de, entre otras ventajas, poder tratar con cambios en dimensiones que cambian lentamente (*slowly-changing dimensions*).

2.4. Posibles herramientas de apoyo

Diversos trabajos apoyan el interés de utilizar herramientas interactivas para el aprendizaje de bases de datos [8, 35, 36, 37, 38, 41, 50]. Como se ha indicado en la Introducción, no se conoce ninguna herramienta gratuita que ofrezca la variedad de funcionalidades necesarias para la docencia en bases de datos, y por eso se ha creado DBDAp.

Existen algunas herramientas gráficas de diseño de diagramas que podrían utilizarse para diseñar esquemas E/R. Entre ellas, podríamos destacar EDGE Diagrammer y Dia:

- La herramienta *EDGE Diagrammer* (<http://www.pacestar.com/edge/index.html>) también puede utilizarse para la realización de distintos tipos de diagramas, y en particular, utilizando una plantilla desarrollada por el profesor Santiago Velilla de la Universidad de Zaragoza, diagramas E/R. El Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza adquirió varias licencias para la utilización de esta herramienta. Además, existen versiones de evaluación gratuitas que pueden descargarse desde la página web de *Pacestar Software*.
- De manera similar, podría utilizarse también la herramienta de código abierto *Dia* (<http://live.gnome.org/Dia>), ya que ofrece una plantilla para la construcción de diagramas E/R.

Sin embargo, el soporte para dibujar esquemas E/R es sólo una de las múltiples funcionalidades que ofrece DBDAp. Además, estas herramientas de dibujo no permiten la comprobación de ningún tipo de error de diseño, ni permiten documentar de forma sencilla los dominios de los atributos o restricciones adicionales que afectan a los datos que no puedan representarse con los elementos notacionales del modelo E/R.

3. Funcionalidades de la herramienta

La herramienta se ha desarrollado utilizando el lenguaje Java e incluye una cantidad importante de funcionalidades que la hacen particularmente atractiva como herramienta docente de apoyo para el aprendizaje del diseño de bases de datos y almacenes de datos. En esta sección se revisan las funcionalidades principales.

3.1. Diseño conceptual de bases de datos

La herramienta sirve de soporte para el diseño de esquemas conceptuales con corrección asistida (como ejemplo, puede verse la figura 2). Dispone de un entorno gráfico para la creación de esquemas utilizando el modelo Entidad/Relación, así como una funcionalidad de detección de errores que proporciona sugerencias de solución. Gracias a la funcionalidad de detección de errores, el alumnado puede aprender a evitar algunos errores comunes básicos y, por tanto, a realizar mejores diseños, comprobando los errores cometidos y entendiendo las posibles soluciones para ellos. Además, es posible definir restricciones en lenguaje natural e información acerca de los dominios de datos utilizados, con lo que todo el diseño conceptual (no sólo el diagrama E/R propiamente dicho) es gestionado y almacenado por la herramienta. También se puede incluir información sobre dependencias funcionales [5] y dependencias multivaluadas [21] que existan entre los datos: esto sirve como mecanismo de documentación de propiedades de los datos y además como etapa previa a la posible aplicación del proceso de normalización sobre el esquema relacional que se obtendrá posteriormente (como se describe en el apartado 3.3.2 y apartado 3.3.3).

Algunos de los elementos soportados en referencia al modelo E/R son los conceptos de entidad y tipo de entidad (tipos de entidad fuerte y tipos de entidad débiles), atributos y dominios, atributos identificadores (primarios y alternativos), interrelación y tipo de interrelación (grado, tipo de correspondencia, roles, atributos de la interrelación) y otras restricciones. Con respecto al modelo E/R extendido se pueden mencionar: tipos de atributos (simples/compuestos, obligatorios/optativos, monovaluados/multivaluados, base/derivados), cardinalidad mínima y máxima, participación (obligatoria/opcional, notación con cardinalidades), interrelaciones exclusivas, generalización/especialización y agregación.

Además, se ha añadido soporte para definir esquemas conceptuales en UML y conversiones automáticas entre ambas opciones de modelado conceptual. Como ejemplo, la parte derecha de la figura 3 muestra el esquema UML obtenido como resultado de la transformación automática del esquema de la figura 2. Puede observarse, entre otras cosas, la sustitución de los tipos de interrelación por asociaciones y la utilización de estereotipos UML especiales, como «PK» (*Primary Key*) para representar claves primarias.

3.2. Uso de notaciones alternativas para el modelo E/R

La herramienta permite la configuración de la notación a utilizar en el esquema E/R, de modo que la notación empleada durante la etapa de diseño conceptual se puede modificar, como se ilustra en la figura 4. Incluso se pueden crear plantillas de notación, que pueden ser utilizadas en otras ocasiones y por cualquier persona que utilice la herramienta. Sobre un esquema E/R creado con la herramienta es posible aplicar distintas plantillas de notación para ver el esquema según diversas convenciones. Esta funcionalidad resulta especialmente útil debido a la no existencia de una notación estándar para el modelo E/R.

3.3. Diseño lógico de bases de datos

Como soporte para el diseño lógico de bases de datos, la herramienta ofrece dos funcionalidades principales: la obtención de esquemas relacionales y la normalización de dichos esquemas.

3.3.1. Obtención de esquemas relacionales

La herramienta permite realizar una transformación automática del esquema conceptual en un esquema relacional (figura 5). Este proceso se puede llevar a cabo opcionalmente paso a paso, ilustrando las distintas fases del proceso de transformación aplicado. En cada paso se muestran explicaciones acerca de cómo se va creando el esquema relacional a partir de los distintos elementos representados en el esquema conceptual. Además, si es preciso, se pregunta al usuario sobre posibles decisiones que se puedan tomar durante el proceso, aconsejando sobre las ventajas/inconvenientes de las distintas alternativas existentes. Esto permite aprender todo el proceso de transformación de un esquema E/R a un esquema relacional y entender las posibles alternativas y decisiones que se pueden tomar a lo largo del proceso.

Se pretende que esta herramienta de traducción paso a paso sea de utilidad para el alumnado, ya que le ayudará a entender algunos aspectos básicos del proceso de traducción y a evitar errores que habitualmente suele cometer en este proceso. Por ejemplo, al transformar una especialización/generalización de un esquema E/R a un esquema relacional hay diversas opciones de traducción posibles, cada una con sus ventajas e inconvenientes: el alumnado suele encontrar este aspecto difícil de llevar a la práctica, ya que habitualmente no sabe qué opción escoger o realiza una traducción insuficiente (por ejemplo, dejando de especificar restricciones importantes, que deberán verificarse mediante disparadores o *triggers* [12] o mecanismos similares). Como ejemplo, la figura 6 muestra un sencillo caso de especialización (parte izquierda de la figura) junto con el diálogo que le aparece al usuario para que elija una opción de traducción (parte derecha de la figura).

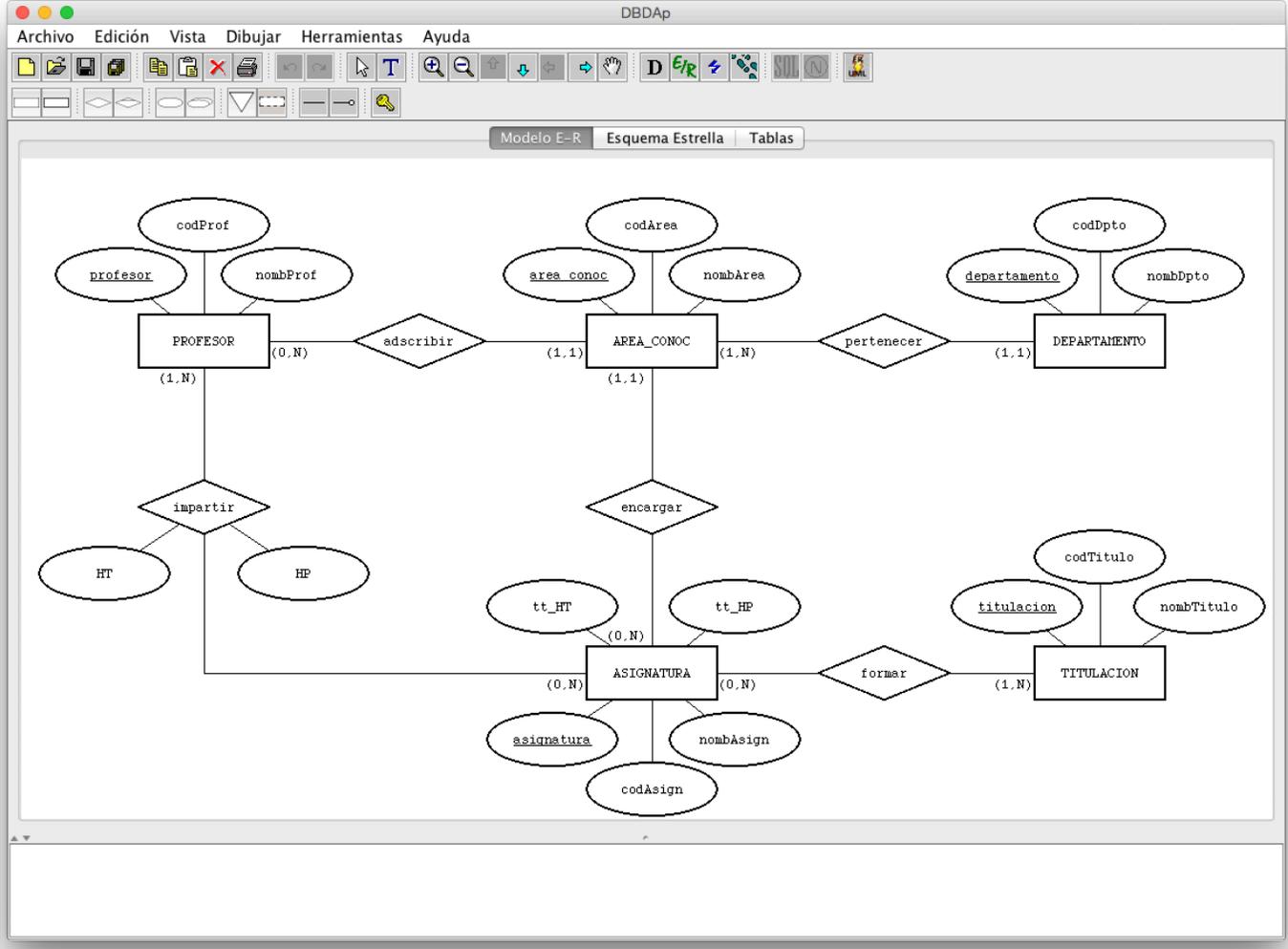


Figura 2: Ejemplo de esquema conceptual utilizando el modelo E/R

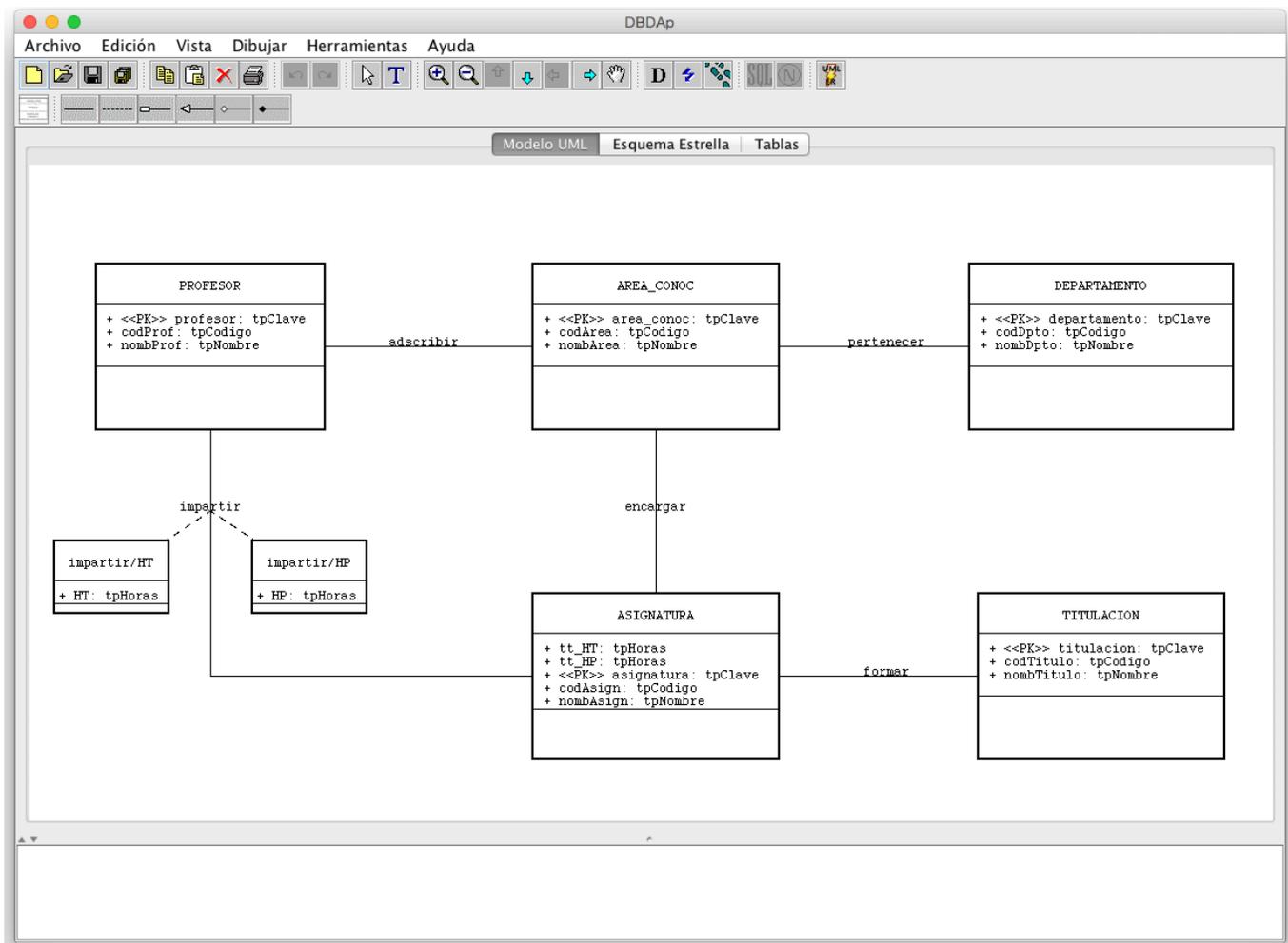


Figura 3: Ejemplo de esquema conceptual utilizando un diagrama de clases UML

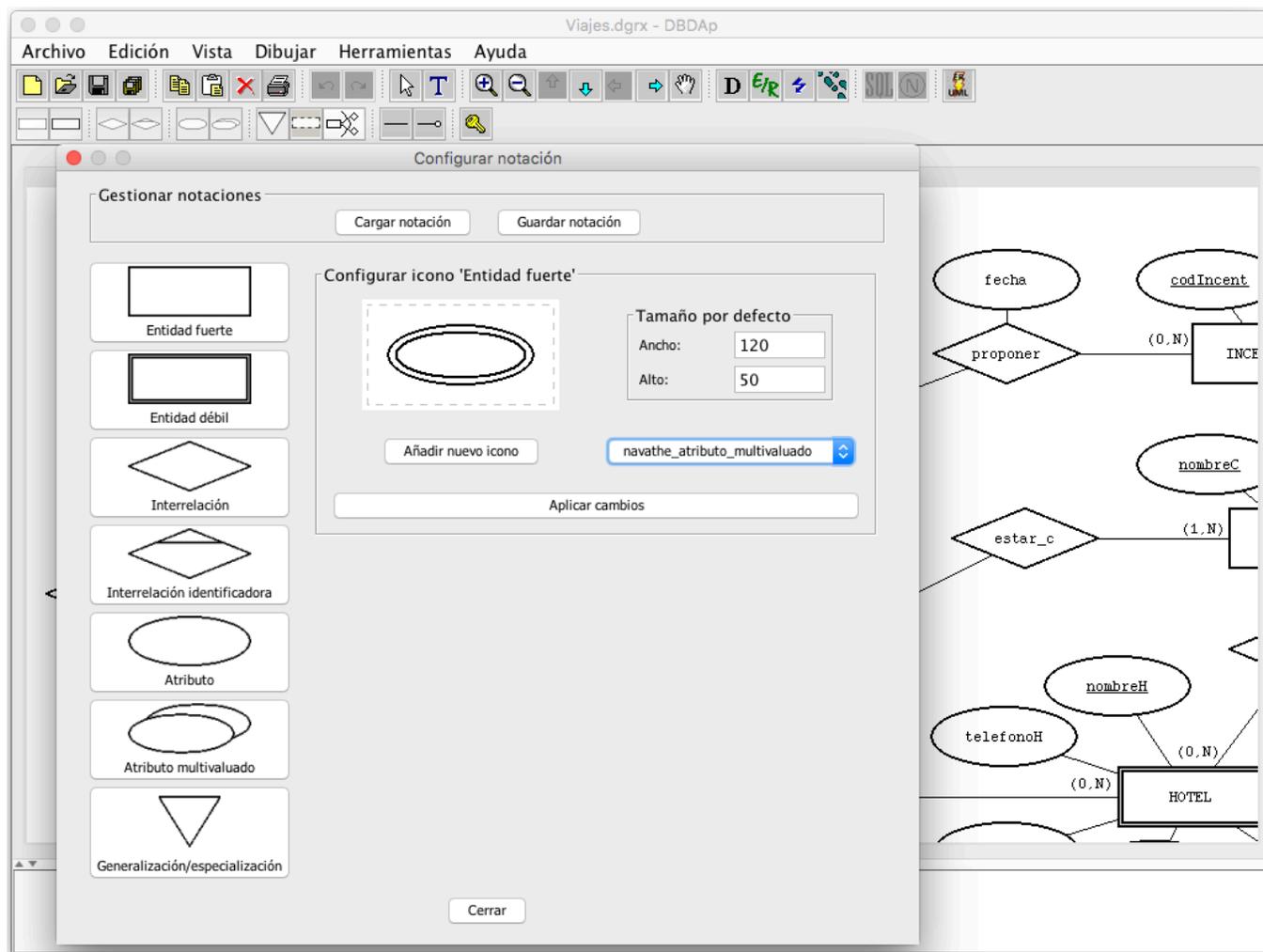


Figura 4: Configuración de la notación a utilizar para el diseño conceptual

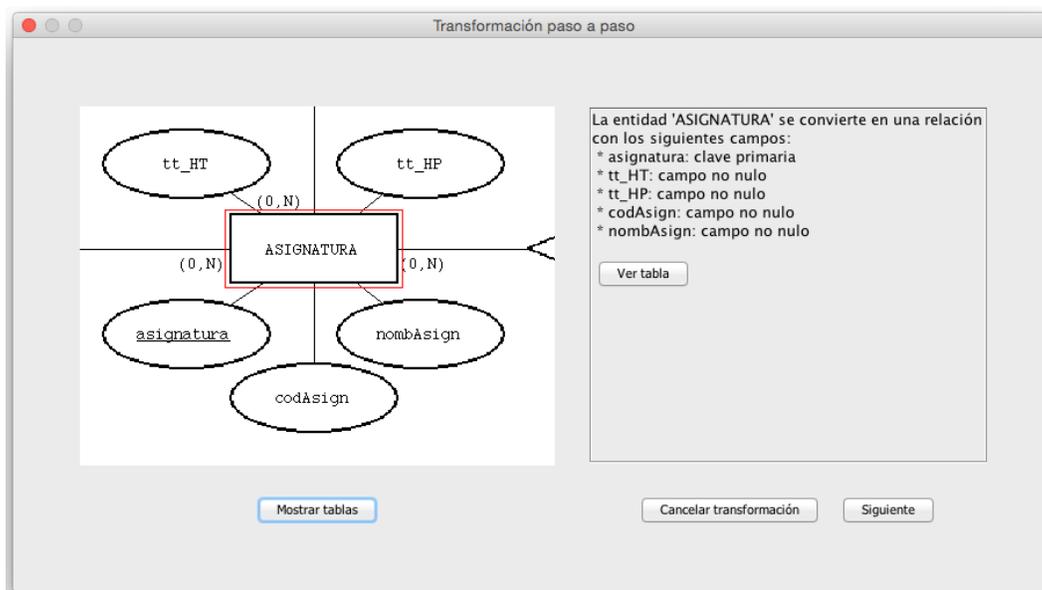


Figura 5: Ejemplo de un paso de transformación del esquema E/R

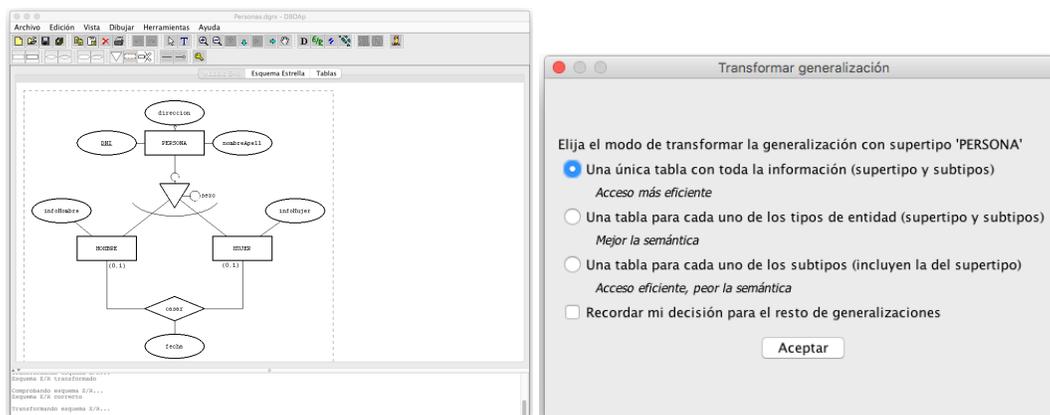


Figura 6: Especialización/generalización: diálogo para elegir un método de transformación a esquema relacional

Otro ejemplo de error típico del alumnado es no percatarse de que al traducir un esquema E/R en un esquema relacional hay una pérdida de semántica que habitualmente requiere la introducción de restricciones adicionales para garantizar las propiedades deseadas de los datos. Un caso habitual es la adición de restricciones de cardinalidad mínima. Este tipo de restricciones son añadidas automáticamente por la herramienta, como puede apreciarse en la figura 7.

Otros errores comunes que suele cometer el alumnado son la definición incorrecta de claves ajenas, la traducción inadecuada de tipos de entidad débil o agregaciones, o el uso incorrecto de las interrelaciones ternarias [16]. La herramienta ayuda al alumnado a evitar y detectar estos errores gracias a sus funcionalidades de traducción automática (comentadas en esta sección) y de normalización de esquemas relacionales (descritas en el apartado 3.3.2).

3.3.2. Normalización de esquemas relacionales

Otra utilidad de la herramienta es la de normalizar esquemas relacionales introducidos manualmente por el usuario o generados por la aplicación, aplicando la teoría de la normalización [26, 46]. Esta utilidad permite al usuario comprobar si un esquema se encuentra normalizado, así como normalizarlo si es necesario. También dispone de un método de normalización guiado, con explicaciones, no excesivamente formales, de los pasos tomados y de la teoría de la normalización aplicada.

Un ejemplo de un paso durante el proceso de normalización guiado se muestra en la figura 8, donde se muestra una etapa del proceso de la normalización donde el usuario tiene que tomar una decisión entre pasar a forma normal de Boyce-Codd [13] y perder una dependencia funcional o quedarse en tercera forma normal [6]; este es un ejemplo de dificultad clásica durante el proceso de normalización, ya que como se indica Beerli y Bernstein [4] no siempre es posible encontrar un esquema en forma normal de Boyce-Codd que represente un conjunto dado de dependencias funcionales. Al concluir el proceso de normalización, el usuario puede guardar un informe/resumen del proceso realizado en formato texto.

3.3.3. Detección automática de dependencias funcionales y multivaluadas

Para llevar a cabo el proceso de normalización es necesario identificar e introducir previamente las dependencias existentes. Como ejemplo, en la figura 9 se ilustra el proceso de introducción manual de dependencias funcionales. Las dependencias multivaluadas se introducen a través de una ventana similar.

Como ayuda para definir las dependencias, la herramienta dispone de una sencilla funcionalidad que permite detectar de forma automática posibles dependencias (tanto funcionales como multivaluadas) a partir de tablas pobladas con un número significativo de tuplas. En el ejemplo de la figura 10

se muestra el resultado de la ejecución del proceso de detección automático de dependencias a partir de un conjunto de datos cargado para un esquema dado. El usuario debe decidir qué dependencias desea incorporar (es decir, debe validar las dependencias descubiertas), ya que lógicamente el hecho de que dichas dependencias se verifiquen en los datos cargados no significa que sean generalizables a cualquier conjunto de datos en el dominio del problema que se está modelando. La implementación de esta funcionalidad puede mejorarse incorporando algoritmos eficientes y robustos para la detección de las dependencias. Papenbrock *et al.* muestran un ejemplo en un artículo reciente donde comparan diversas aproximaciones para el descubrimiento de dependencias funcionales [40].

3.4. Generación de código SQL

La herramienta soporta una traducción automática del esquema relacional obtenido, o del esquema en estrella diseñado, a lenguaje SQL (*Structured Query Language*) para distintos Sistemas de Gestión de Bases de Datos (SGBD). Para ello, genera un archivo de órdenes (*script*) que se puede ejecutar directamente sobre el SGBD. El usuario puede configurar la forma de generar el archivo de órdenes, estableciendo los tipos de datos correspondientes al SGBD donde se pretende utilizar. En el prototipo actual, se han considerado Access, Oracle y MySQL. No obstante, la herramienta se podría configurar para ser utilizada con otros SGBD. Como ejemplo de expresión SQL generada, puede verse la figura 11.

3.5. Diseño de esquemas en estrella

La herramienta permite la definición de esquemas lógicos en estrella [1] y su transformación automática a SQL. La notación utilizada para la definición de esquemas en estrella está inspirada por la propuesta por Ralph Kimball [28, 30], considerado una de las dos personas más influyentes en lo referente al diseño de almacenes de datos junto a William H. Inmon.

Como ejemplo, la figura 12 muestra un esquema en estrella correspondiente al proceso de negocio de ventas de una empresa. De acuerdo con la metodología de Kimball, el *grano* del proceso de negocio en este caso es la venta de un producto en una fecha concreta a un cliente determinado. La tabla de hechos central (*Ventas*) contiene como *métricas* (medidas de negocio o hechos) para el análisis el número de unidades vendidas de ese producto (*unidades*) y el precio total de venta (*precioTotal*). Además, el número de ticket de venta (*numTicket*) permite agrupar todas los productos vendidos a un cliente en la misma venta, jugando el papel de *dimensión degenerada* (*DD*). Como *dimensiones* primarias, tenemos *Producto*, *Fecha* y *Cliente*, mientras que *Promoción* y *Almacén* son dimensiones secundarias (caracterizan las medidas tomadas pero no son fundamentales para identificar los hechos). Nótese que la clave primaria (*PK*) de *Ventas* es una clave compuesta por *idProducto*, *idFecha*, e *idCliente* (claves ajenas que apuntan a las correspondientes tablas de dimensión primaria). Alternativamente, podría haberse definido una clave primaria com-

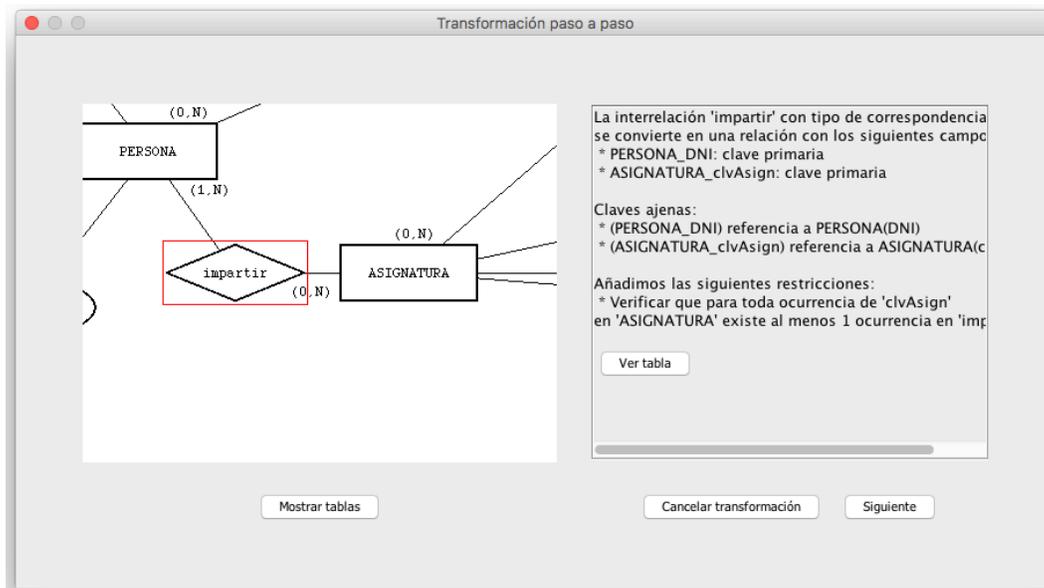


Figura 7: Ejemplo de un paso de transformación del esquema E/R que muestra la adición automática de restricciones necesarias

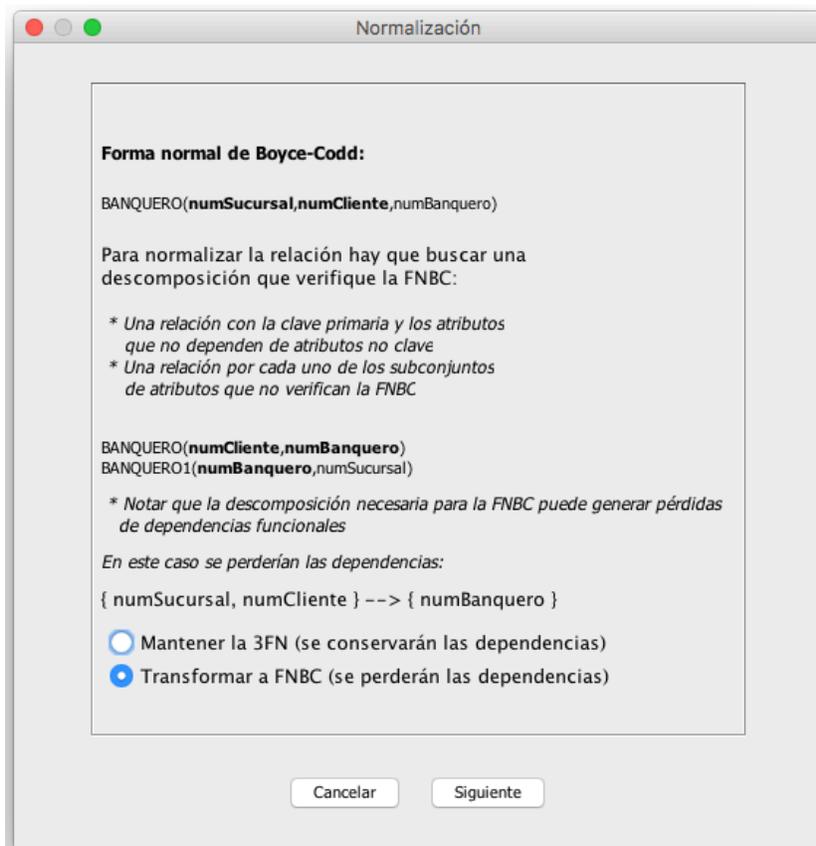


Figura 8: Ejemplo de un paso de normalización de un esquema relacional

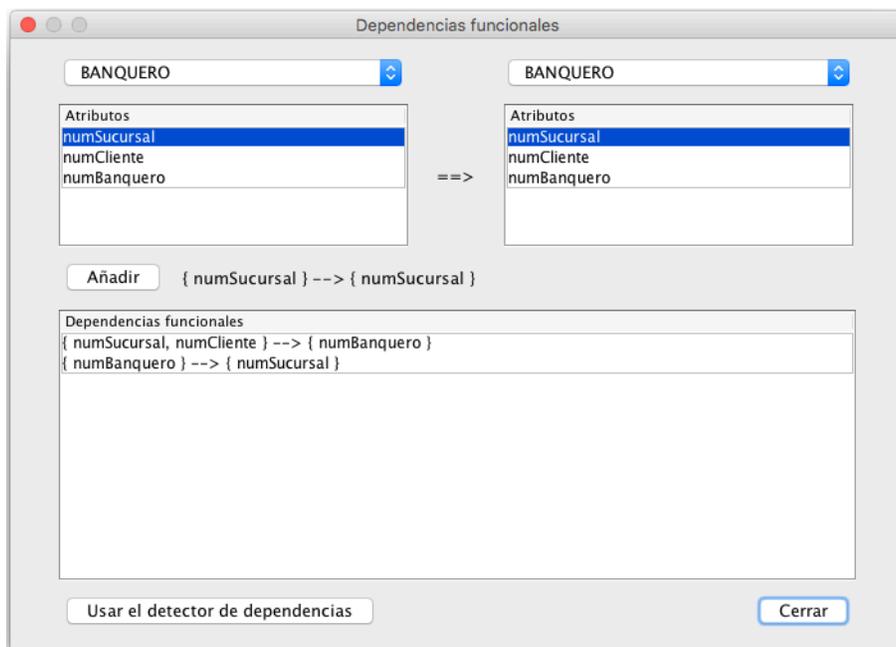


Figura 9: Introducción manual de dependencias funcionales

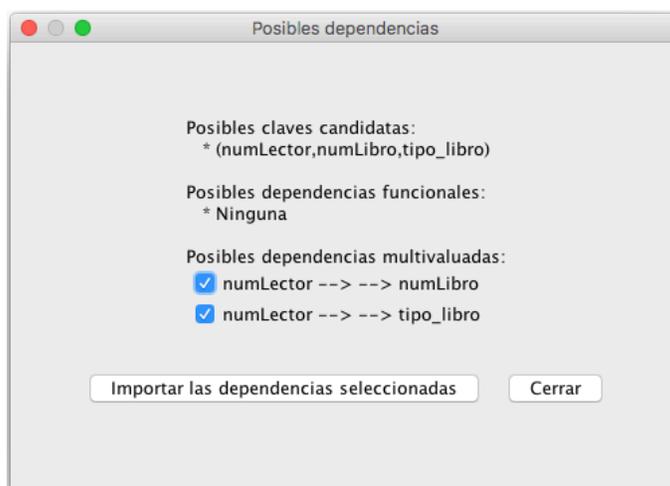


Figura 10: Detección automática de dependencias funcionales

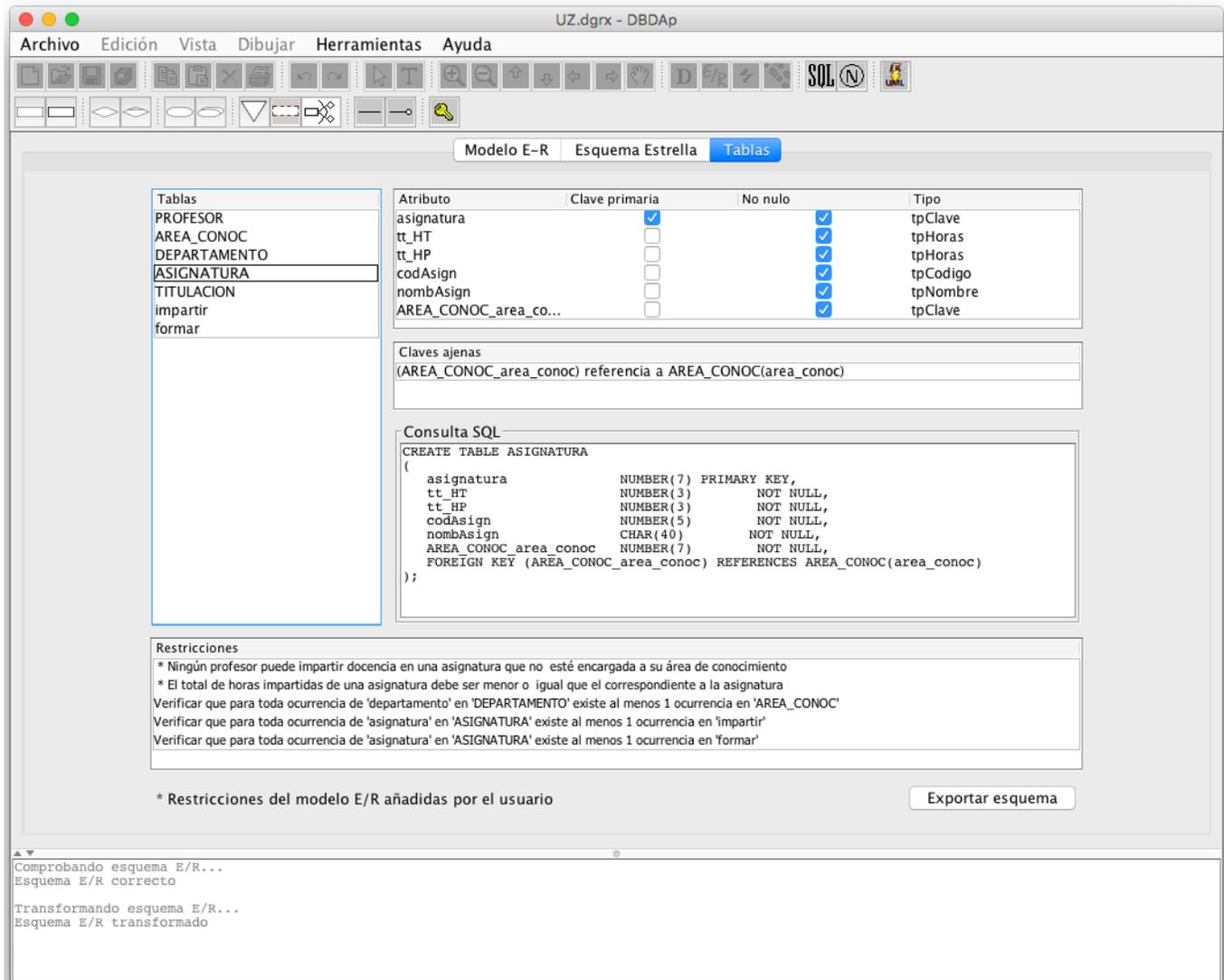


Figura 11: Ejemplo de expresión SQL para crear una relación del esquema obtenido

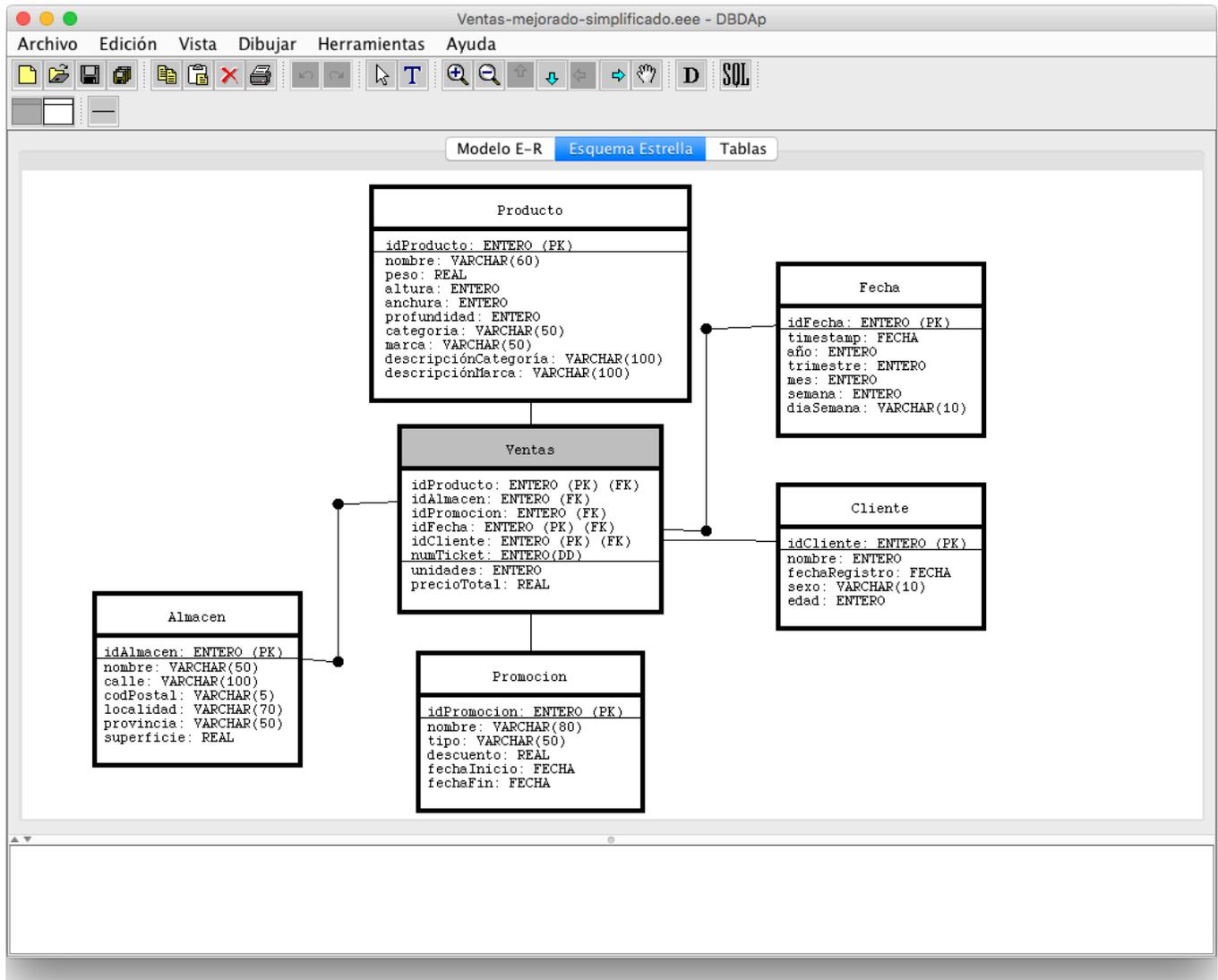


Figura 12: Ejemplo de esquema en estrella definido con la herramienta

puesta por la dimensión degenerada *numTicket* y el atributo *idFecha*, si asumimos que el número de ticket es único para cada día.

El esquema en estrella anterior es un esquema puro, ya que consta de una tabla de hechos central y tablas de dimensiones alrededor de dicha tabla. Sin embargo, la herramienta permite también definir esquemas con *copos de nieve*, como se muestra en la figura 13. En dicha figura, podemos ver cómo la información acerca de las categorías y marcas de los productos se ha incluido en tablas aparte. En general, este tipo de esquemas, que normalizan el esquema original, están desaconsejados, pero son permisibles y pueden resultar de interés en algunas ocasiones. Por otro lado, en este nuevo esquema se ha decidido incluir una *minidimensión (RangoEdad)* para representar el rango de edad de los clientes en lugar de la edad concreta de los mismos; esta es una técnica habitual para gestionar las denominadas *rapidly changing monster dimensions*. Aunque no se muestra en esta figura, los atributos *fechaInicio* y *fechaFin* de la tabla de dimensión *Promoción* también se pueden definir como claves ajenas que referencian a la tabla de dimensión *Fecha*: serían entonces ejemplos de *outriggers* permisibles según las guías de diseño de Ralph Kimball.

3.6. Registro de la información generada

La herramienta incorpora diversas utilidades fáciles de usar para guardar los esquemas definidos y los resultados generados, exportándolos como ficheros de texto, imprimiendo los esquemas generados o guardándolos como imágenes. Un ejemplo de situación donde esto puede resultar de utilidad es tras ejecutar el proceso de normalización o después de realizar la transformación del esquema E/R en esquema relacional. En la figura 14 muestra el código SQL generado a partir del esquema en estrella de la figura 12.

Por supuesto, los esquemas definidos pueden ser guardados e intercambiados entre el alumnado, facilitando la posible realización de actividades de evaluación entre pares. El interés de la evaluación entre pares se ha enfatizado para el aprendizaje en general [52, 54] y en el contexto de la Informática [22, 31, 45, 53] y bases de datos [34] y sistemas de información [45] en particular. Por el momento, no se ha utilizado DBDAp como soporte para la realización de este tipo de actividades de evaluación entre pares para la valoración de diseños entre estudiantes, pero el hecho de que los diseños se almacenen en un fichero, y con una notación consistente, facilitaría el intercambio de los esquemas entre estudiantes, siendo mucho mejor que otras alternativas que podrían implicar el intercambio de soluciones realizadas a mano en papel.

4. Evaluación de DBDAp

La herramienta DBDAp está a disposición del profesorado y alumnado de asignaturas relacionadas con bases de datos y almacenes de datos, sirviendo de soporte para la realización de diversas tareas, como la definición de esquemas de bases de

datos y el planteamiento de ejercicios y borradores de soluciones. La herramienta DBDAp se ha ido mejorando a lo largo de los años y continúa en un proceso de mejora continua, ya que se van incorporando sugerencias e ideas que van surgiendo. No obstante, actualmente se considera que ha alcanzado un grado de madurez suficiente como para su utilización en la docencia.

La herramienta se ha desarrollado considerando principalmente las necesidades docentes de varias asignaturas relacionadas con la titulación del Grado en Ingeniería en Informática de la Universidad de Zaragoza: «Bases de datos», «Bases de datos 2», y «Almacenes y minería de datos» (descritas con mayor detalle en el anexo A). Además, también se ha utilizado en la asignatura «Manipulación y análisis de grandes volúmenes de datos» del Máster universitario en Ingeniería de sistemas e Informática de la Universidad de Zaragoza. La utilización de la herramienta se ha planteado por el momento en dichas asignaturas como algo opcional, tanto para el alumnado como para el profesorado.

4.1. Percepción del profesorado

A través de la experiencia personal impartiendo docencia en dichas asignaturas, se ha observado que la herramienta resulta de gran utilidad para varias actividades importantes.

La primera actividad es la *realización de prácticas por parte del alumnado*. DBDAp ofrece soporte para la realización de los diseños de bases de datos y almacenes de datos, permitiendo diseñar los esquemas de forma cómoda y pudiendo utilizar, entre otras alternativas, la notación explicada en clase. Además, también es posible explotar funcionalidades de la herramienta que resultan de interés en este contexto, como la transformación automática de esquemas E/R a esquemas relacionales o el soporte para la normalización de esquemas relacionales.

La segunda es el *estudio por parte del alumnado*. DBDAp ofrece funcionalidades que ayudan al alumnado a entender mejor diversos aspectos relacionados con el diseño y que habitualmente tiene dificultades para interiorizar. En este sentido, resultan de gran utilidad, entre otras, las siguientes funcionalidades:

- *La funcionalidad de transformación paso a paso de esquemas E/R a esquemas relacionales*. Utilizando dicha funcionalidad, el alumnado puede entender fácilmente cómo cada uno de los elementos del esquema E/R se transforma al pasar a un esquema relacional. Puede ver, por ejemplo, cómo se traducen abstracciones con las que suelen tener más dificultades, como los tipos de entidad débil, los tipos de entidad agregados, las especializaciones/generalizaciones, o los tipos de interrelación ternarias. Además, gracias a esta funcionalidad aprende dónde pueden aparecer decisiones de diseño donde el diseñador debe intervenir (típicamente, en referencia a las distintas formas posibles de traducir una especialización/generalización, cada una con sus ventajas e incon-

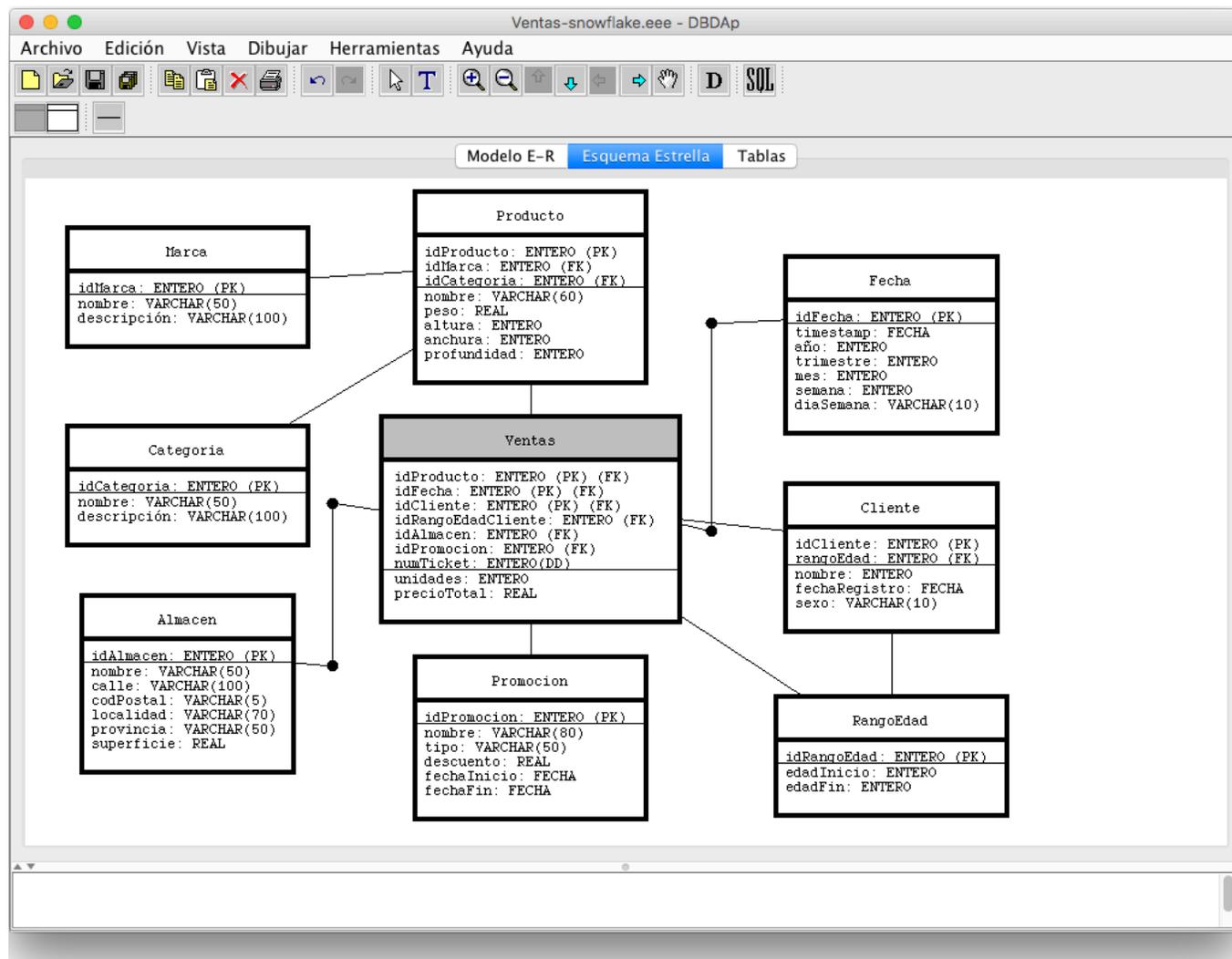


Figura 13: Ejemplo de esquema en copo de nieve definido con la herramienta

```

1 CREATE TABLE Ventas
2 (
3     idProducto          NUMBER,
4     idAlmacen           NUMBER      NOT NULL,
5     idPromocion         NUMBER      NOT NULL,
6     idFecha             NUMBER,
7     idCliente           NUMBER,
8     unidades            NUMBER      NOT NULL,
9     precioTotal         FLOAT        NOT NULL,
10    numTicket            INTEGER     NOT NULL,
11    PRIMARY KEY(idProducto, idFecha, idCliente),
12    UNIQUE(numTicket, idFecha),
13    FOREIGN KEY(idProducto) REFERENCES Producto (idProducto),
14    FOREIGN KEY(idAlmacen) REFERENCES Almacen (idAlmacen),
15    FOREIGN KEY(idPromocion) REFERENCES Promocion (idPromocion),
16    FOREIGN KEY(idFecha) REFERENCES Fecha (idFecha),
17    FOREIGN KEY(idCliente) REFERENCES Cliente (idCliente));
18
19 CREATE TABLE Producto
20 (
21     idProducto          NUMBER,
22     nombre              CHAR(60)    NOT NULL,
23     peso                FLOAT        NOT NULL,
24     altura              NUMBER      NOT NULL,
25     anchura             NUMBER      NOT NULL,
26     profundidad        NUMBER      NOT NULL,
27     categoria           CHAR(50)    NOT NULL,
28     marca               CHAR(50)    NOT NULL,
29     descripciónCategoría CHAR(100) NOT NULL,
30     descripciónMarca    CHAR(100)  NOT NULL,
31     PRIMARY KEY(idProducto));
32
33
34 CREATE TABLE Almacen
35 (
36     idAlmacen           NUMBER,
37     nombre              CHAR(50)    NOT NULL,
38     calle               CHAR(100)   NOT NULL,
39     codPostal           CHAR(5)     NOT NULL,
40     localidad           CHAR(70)    NOT NULL,
41     provincia           CHAR(50)    NOT NULL,
42     superficie          FLOAT        NOT NULL,
43     PRIMARY KEY(idAlmacen));
44
45

```

(root) > Users > silarri > Desktop > Ventas.sql | Latin-1 | SQL | Ln: 1 Col: 1

Figura 14: Ejemplo de fragmento de código SQL correspondiente a un esquema en estrella

venientes). Por último, otro de los aspectos clave donde la transformación paso a paso ayuda al aprendizaje es en todo lo referente a las restricciones que en ocasiones es necesario añadir al pasar de un esquema E/R a un esquema relacional; ejemplos de esto último son las restricciones necesarias para exigir en el esquema relacional las cardinalidades mínimas presentes en el esquema E/R o restricciones relacionadas con la especializaciones/generalizaciones presentes en el esquema E/R, que deben añadirse en el esquema relacional correspondiente para tratar de salvar esa pérdida semántica que se produce al transformar una jerarquía en tablas planas.

- *La funcionalidad de normalización de esquemas relacionales.* Gracias al soporte para la normalización de esquemas relacionales, el alumnado puede experimentar con distintos esquemas relacionales y jugar con datos de ejemplo para entender el proceso de normalización y cómo ir aplicándolo paso a paso para pasar de una forma normal a otra superior. Esto se considera de interés porque la teoría de la normalización es considerada habitualmente por el alumnado como una parte difícil de las asignaturas de bases de datos y con frecuencia encuentran problemas para aplicarla correctamente.
- *La funcionalidad de generación automática de código SQL.* El alumnado puede definir su esquema y automáticamente obtener código SQL que le permitirá implementarlo sobre un SGBD relacional y probarlo. Por tanto, puede realizar un diseño, implementarlo e introducir datos, y luego formular consultas sobre él para asegurarse de que el diseño permite satisfacer las necesidades de información existentes para resolver un problema dado.
- *Las funcionalidades de soporte para el diseño de esquemas.* Diseñando esquemas con la herramienta, el alumnado puede asegurarse de que está utilizando una notación correcta y aprender de ello. Por ejemplo, DBDAp no permite definir un atributo de un tipo de interrelación como clave. Del mismo modo, no permite la definición de un atributo como clave primaria en un subtipo, sino únicamente como clave alternativa o atributo con valores únicos (el subtipo hereda la clave primaria del supertipo). Estos son errores bastante comunes entre el alumnado, a pesar de que es algo que no permite el modelo relacional

Una tercera actividad para el que DBDAp resulta útil es la *preparación de exámenes y ejercicios por parte del profesorado*. Con DBDAp es posible definir gráficamente esquemas adecuados para un enunciado de ejercicio planteado, utilizar sus funcionalidades para realizar algunas transformaciones y verificaciones automáticas, y finalmente obtener un esquema que se puede implementar directamente en un SGBD relacional para hacer pruebas.

Y, finalmente, también es útil para la *realización de ejemplos por parte del profesorado*. DBDAp permite obtener esquemas estéticamente agradables con relativamente poco esfuerzo. Por tanto, es una herramienta adecuada para realizar ejemplos en clase y para diseñar esquemas de solución que puedan distribuirse al alumnado como documentación adicional. La funcionalidad de transformación paso a paso de esquemas E/R a esquemas relacionales también resulta de gran interés para la docencia. En cursos previos, cuando DBDAp todavía no estaba disponible, se realizó algún ejemplo de transformación paso a paso para el alumnado utilizando transparencias PowerPoint con animaciones, con el esfuerzo de desarrollo que ello supone para el profesorado. Con DBDAp, la obtención de estos ejemplos paso a paso es inmediata.

Por todo ello, se considera que la herramienta ha tenido un efecto positivo tanto para la docencia como el aprendizaje del diseño de bases de datos y almacenes de datos. Aunque resulta muy complicado cuantificar el impacto de la herramienta, parece claro que: 1) al profesorado le permite un ahorro de tiempo importante en la preparación de ejemplos y la posibilidad de realizar diseños estéticamente agradables; y 2) al alumnado le sirve de soporte para la realización de ejercicios y prácticas y con su utilización puede aprender a evitar errores comunes. Antes del desarrollo de DBDAp, no se disponía de una herramienta de apoyo con funcionalidades comparables, por lo que debían utilizarse herramientas de diseño que no estaban especializadas para bases de datos ni ofrecían una ayuda enfocada en dicho dominio.

4.2. Percepción del alumnado

Con objeto de recopilar información acerca de su experiencia con la herramienta y su opinión, en marzo/abril de 2016 se realizó una encuesta entre el alumnado del curso 2015/2016 de «Bases de datos 2» y otra entre el alumnado de «Manipulación y análisis de grandes volúmenes de datos». Además, en octubre de 2016 se realizó otra encuesta complementaria entre el alumnado del curso 2016/17 de «Almacenes y minería de datos». Las tres asignaturas son impartidas en la Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza. Las encuestas se dirigieron a todo el alumnado de dichas asignaturas e incluían preguntas cerradas (sí/no), preguntas de respuesta múltiple, preguntas de valoración en una escala de 1 a 10 y preguntas abiertas para recoger realimentación que el alumnado quiera proporcionar expresándola de forma libre. Las encuestas se rellenaron a través de la plataforma Moodle de la universidad. El contenido concreto de las preguntas puede verse en los Anexos B y C de este artículo. A continuación, analizamos los resultados más llamativos extraídos de las respuestas a dichas encuestas.

4.2.1. Resultados de la encuesta al alumnado de «Bases de datos 2»

La encuesta al alumnado de «Bases de datos 2» tenía carácter voluntario y fue contestada por 25 estudiantes de 42,

lo que representa el 60 % del total. Las conclusiones más relevantes de los resultados obtenidos al analizar las respuestas proporcionadas por el alumnado son las siguientes:

- La mayor parte del alumnado (un 92 %) está satisfecho con la aplicación DBDAp, valorando en media la aplicación globalmente con un 7,2, y sus funcionalidades y facilidad de uso con un 7,3.
- La mayor parte del alumnado (un 96 %) considera de interés seguir mejorando y ampliando la aplicación DBDAp.
- Las funcionalidades más utilizadas son: el diseño conceptual utilizando esquemas E/R (un 96 % del alumnado), la traducción de esquemas E/R en esquemas relacionales (un 68 %), la generación automática de código SQL (un 40 %), la traducción de esquemas E/R en esquemas relacionales utilizando el modo de transformación paso a paso (un 32 %) y la normalización de esquemas relacionales (un 24 %). Funcionalidades más peculiares, como el diseño conceptual utilizando diagramas de clase UML, la detección automática de dependencias funcionales y multivaluadas a partir de un conjunto de datos dado, y la definición de notaciones alternativas para el modelo E/R, también fueron utilizadas por un 16 %, 12 %, y 8 % del alumnado, respectivamente. La funcionalidad de diseño de esquemas en estrella no fue utilizada por ningún alumno, ya que los esquemas en estrella no forman parte del programa de «Bases de datos 2».
- Ninguno de los alumnos conoce ninguna aplicación de apoyo al aprendizaje o diseño de bases de datos mejor que DBDAp o que la supere en alguna funcionalidad. Únicamente señalan, como aplicaciones parcialmente relacionadas, soluciones como MySQL Workbench, Dia, EDGE Diagrammer, draw.io, y Modelio (UML), básicamente para las tareas de definición de esquemas.

Además, entre las mejoras que sugieren, podemos destacar: soporte para el modelo objeto/relacional, generación automática de datos aleatorios de prueba, herramientas de importación y exportación para integrar con otras aplicaciones de dibujo y añadir opciones para deshacer la última acción y para rehacer, en su caso, la acción deshecha.

4.2.2. Resultados de la encuesta al alumnado de «Manipulación y análisis de grandes volúmenes de datos»

La encuesta al alumnado de «Manipulación y análisis de grandes volúmenes de datos» tenía carácter voluntario y fue contestada por 6 estudiantes de 7, lo que representa el 86 % del total. Las conclusiones más relevantes de los resultados obtenidos al analizar las respuestas proporcionadas por el alumnado son las siguientes:

- El alumnado en su totalidad (un 100 %) está satisfecho con la aplicación DBDAp, valorando en media la aplicación globalmente con un 8,0, sus funcionalidades con un 7,8, y su facilidad de uso con un 7,7.
- El alumnado en su totalidad (un 100 %) considera de interés seguir mejorando y ampliando la aplicación DBDAp.
- Las funcionalidades más utilizadas son: el diseño de almacenes de datos (esquemas en estrella), la generación automática de código SQL (en ambos casos, un 83 % de los alumnos) y la definición de notaciones alternativas para el modelo E/R (un 33 %). Como funcionalidades menos utilizadas por este grupo de alumnos, encontramos el diseño conceptual utilizando esquemas E/R, la traducción de esquemas E/R en esquemas relacionales y la detección automática de dependencias funcionales y multivaluadas a partir de un conjunto de datos dado (un 17 % en todos los casos). Los porcentajes observados son consistentes con el hecho de que el diseño de almacenes de datos es la tarea fundamental realizada en la asignatura «Manipulación y análisis de grandes volúmenes de datos» en lo que concierne al almacenamiento de grandes volúmenes de datos.
- Ninguno de los alumnos conoce ninguna aplicación de apoyo al aprendizaje o diseño de bases de datos mejor que DBDAp o que la supere en alguna funcionalidad. Únicamente señalan, como aplicaciones parcialmente relacionadas, soluciones como MySQL Workbench, Dia, Modelio, Papyrus, PHPMyAdmin, y Astah, pero aclaran que las han utilizado únicamente para dibujar los diagramas.

Además, entre las mejoras que sugieren, se centran sobre todo en funcionalidades que permitirían una utilización más intuitiva y sencilla de la herramienta.

4.2.3. Resultados de la encuesta al alumnado de «Almacenes y minería de datos»

La encuesta al alumnado de «Almacenes y minería de datos» se solicitaba como parte de la entrega de una práctica (primera práctica de la asignatura) y fue contestada por 19 estudiantes de 20, lo que representa el 95 % del total. Las conclusiones más relevantes de los resultados obtenidos al analizar las respuestas proporcionadas por el alumnado son las siguientes:

- El alumnado considera, en general, que DBDAp es una herramienta de interés en asignaturas relacionadas con bases de datos (valoración media de 7,3 sobre 10) y también en el contexto de la asignatura (valoración media de 7,5 sobre 10), que es una buena herramienta como soporte para el diseño de bases de datos y la realización de ejercicios (valoración media de 7,3 sobre 10),

que facilita el diseño de bases de datos y almacenes de datos (valoración media de 7,6 sobre 10), y que es fácil de manejar (valoración media de 7,2 sobre 10).

- Asimismo, el alumnado considera de interés seguir manteniendo la herramienta (valoración media de 7,8 sobre 10) y ampliándola (valoración media de 8,4 sobre 10).
- Si bien el alumnado claramente considera que la herramienta ayuda a realizar los diseños, el componente puramente didáctico se ha valorado en menor medida. Así, el alumnado ha valorado con una valoración media de 6,2 sobre 10 su conformidad con la frase «DBDAp me ayuda en el aprendizaje de diversos aspectos relacionados con asignaturas de bases de datos». Probablemente esto se deba, en parte, a que todavía no se han incorporado orientaciones didácticas de soporte para algunas funcionalidades, como por ejemplo en lo referente al diseño de almacenes de datos.

En cuanto a qué aspectos les permite aprender mejor la herramienta, entre las respuestas destacan la transformación de esquemas entidad/relación a relacional y en general todo lo referente a la definición de esquemas.

5. Conclusiones y trabajo futuro

Este artículo presenta una herramienta que soporta el diseño de bases de datos y almacenes de datos. Además de como posible herramienta de apoyo para la implantación de bases de datos o almacenes de datos, puede destacarse su orientación didáctica. La versión actual de la herramienta está disponible en <http://webdiis.unizar.es/~silarri/DBDAp.html> y una versión previa de la misma fue presentada en la sesión de demostraciones de las XX Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2015) [25]. Se considera una posible futura liberación de su código fuente para facilitar futuras extensiones por parte de la comunidad de usuarios.

El desarrollo de la herramienta es continuo. Actualmente se están incorporando diversas mejoras para facilitar la interacción con el usuario y se están realizando diversos ajustes para adecuar mejor la herramienta a las necesidades docentes. También se dispone de un prototipo preliminar de la herramienta para dispositivos Android. Como trabajo futuro, nos gustaría mejorar el soporte actual para el diseño de almacenes de datos incorporando de forma explícita diversos conceptos de los esquemas en estrella (minidimensiones, superdimensiones, dimensiones basura, tablas puente, etc.); aunque estos elementos pueden representarse con la herramienta, hasta el momento no se ha incorporado dicha terminología ni orientaciones o consejos docentes en referencia a su utilización. Del mismo modo, resulta de interés añadir soporte para aplicar explícitamente la metodología de desarrollo de Kimball,

y para representar jerarquías de atributos que pueden guiar posibles operaciones de *roll-up* y *drill-down* [23] cuando se explote el almacén de datos. Finalmente, pueden incorporarse explicaciones y sugerencias didácticas de apoyo al diseño de esquemas en estrella, entre otras posibles mejoras.

Agradecimientos

Sergio González y Carlos Arias, antiguos estudiantes de la Universidad de Zaragoza, realizaron una excelente contribución en el desarrollo de esta herramienta. El ejemplo de esquema E/R mostrado en la figura 2 fue diseñado por el profesor Santiago Velilla de la Universidad de Zaragoza.

A. Detalles del contexto académico

La herramienta se ha desarrollado en la Escuela de Ingeniería y Arquitectura (EINA), anteriormente Centro Politécnico Superior (CPS)¹, de la Universidad de Zaragoza. En su desarrollo, que se inició aproximadamente a comienzos del año 2009, han participado en distintas etapas dos estudiantes de dicha universidad, que implementaron diversos módulos de la misma como parte de su Proyecto Fin de Carrera en la titulación de Ingeniería en Informática de la Universidad de Zaragoza, bajo la dirección del autor de este trabajo. La titulación de Ingeniería en Informática está actualmente en extinción, habiendo sido reemplazada por el Título Oficial de Graduado o Graduada en Ingeniería Informática de la Universidad de Zaragoza.

En el momento en el que se inició el desarrollo de la aplicación, se pretendía que la herramienta sirviera de apoyo para la docencia de las asignaturas «Ficheros y bases de datos» (asignatura troncal de segundo cuatrimestre, segundo curso, que constaba de 7,5 créditos –6,4 ECTS–) y «Diseño de bases de datos relacionales» (asignatura optativa del cuatrimestre de otoño, que constaba de 6 créditos –5,1 ECTS–) de la titulación de Ingeniería en Informática. Con un planteamiento inicial relativamente modesto, se han ido incorporando a la aplicación una gran variedad de funcionalidades que la hacen atractiva para distintas asignaturas de los planes de estudio actuales. En concreto, en el nuevo grado resulta de utilidad para las asignaturas «Bases de datos», «Bases de datos 2» y «Almacenes y minería de datos». A continuación se comentan los aspectos principales referentes al contenido de dichas asignaturas.

A.1. Asignatura «Bases de datos»

La asignatura «Bases de datos» (ver cuadro 1) tiene como objetivo principal para el alumnado conocer las principales ventajas que aporta el uso de la tecnología de bases de datos en la implementación de sistemas de información, así como familiarizarse con el diseño, implementación, explotación, y

¹Con fecha 15/04/2011 se crea la Escuela de Ingeniería y Arquitectura (EINA), producto de la integración del Centro Politécnico Superior (CPS) y de la Escuela Universitaria de Ingeniería Técnica Industrial (EUITI).

administración de bases de datos. En la asignatura, el alumnado se familiariza con las técnicas más utilizadas para el diseño y gestión de bases de datos. Se tratan los principios básicos del diseño conceptual de bases de datos y del diseño de bases de datos relacionales. También se aborda la administración y explotación de bases de datos, y se introducen algunos conceptos más avanzados que son desarrollados en profundidad en asignaturas posteriores sobre esta temática.

En mayor detalle, la asignatura aborda los temas fundamentales de bases de datos siguiendo un orden lógico. La primera lección introduce el concepto de base de datos y de sistema gestor de base de datos. A continuación se abordan, en orden cronológico, las distintas etapas de diseño. Así, la segunda lección se centra en el diseño conceptual, la tercera lección en el diseño lógico e implementación y la cuarta lección en el diseño físico. Finalmente, la quinta lección se centra en la explotación de bases de datos, incluyendo técnicas de control de concurrencia (fundamental en bases de datos, ya que son multi-usuario por naturaleza), el acceso a bases de datos desde un lenguaje de programación, la gestión de usuarios y otros aspectos básicos de administración.

En cuanto a la presentación de los aspectos más relevantes para el diseño de bases de datos, se sigue una secuenciación cronológica natural, abordando primero el diseño conceptual y posteriormente el diseño lógico (donde se consideran todos los aspectos referentes a las bases de datos relacionales, incluyendo el lenguaje de consultas SQL y la teoría de la normalización) y el diseño físico. Si bien hay cursos y libros que presentan primero el diseño lógico y después el diseño conceptual, probablemente con objeto de dar desde el comienzo una idea más concreta de cuál va a ser la representación final, en la asignatura se prefiere seguir un orden de presentación que se corresponde con el orden en el que se deberían llevar a cabo las distintas etapas del diseño de bases de datos (tal y como se presentó en la figura 1).

Podemos resumir el propósito de la asignatura indicando que el alumnado debe ser capaz de diseñar e implementar las bases de datos necesarias para almacenar y gestionar los datos que se necesiten en los entornos de trabajo más habituales en los que desarrolla su actividad un/a Ingeniero/a en Informática.

A.2. Asignatura «Bases de datos 2»

La asignatura «Bases de datos 2» (ver cuadro 2) tiene como objetivo principal profundizar en los aspectos de diseño y gestión de bases de datos introducidos en la asignatura previa «Bases de datos», así como introducir nuevos elementos referentes a bases de datos objeto-relacionales, bases de datos distribuidas, características de los SGBD disponibles y administración de bases de datos.

En mayor detalle, la asignatura aborda temas avanzados de bases de datos. La primera lección profundiza en el diseño de bases de datos relacionales desde un punto de vista práctico, enfatizando aquellos aspectos que habitualmente plantean dificultades al alumnado, tales como los tipos de interrelación

ternarios, los tipos de entidad débil, las agregaciones y las relaciones de generalización/especialización; de forma similar, en relación con la teoría de la normalización se insiste especialmente en la cuarta y quinta formas normales. Además, se introducen las bases de datos orientadas a objeto y los elementos de orientación a objetos de las bases de datos objeto-relacionales. A continuación, la segunda lección se centra en bases de datos distribuidas, presentando las distintas arquitecturas existentes y las metodologías de diseño consideradas (*top-down* y *bottom-up*). En la tercera lección se abordan aspectos más tecnológicos referentes a los distintos SGBD existentes, incluyendo las *bases de datos NoSQL* [10, 32, 47, 48], analizando sus características y los elementos a valorar al realizar la selección de la tecnología apropiada. Por último, la cuarta lección considera aspectos más avanzados de administración y explotación de una base de datos, por ejemplo mecanismos de seguridad y características concretas de distintos SGBD.

Podemos resumir el propósito de la asignatura indicando que se centra en aspectos avanzados de bases de datos, incluyendo su diseño, administración y gestión en entornos distribuidos, desarrollando en el alumnado las habilidades necesarias para el diseño, creación y gestión de bases de datos de tamaño medio, así como la elección de la tecnología adecuada.

A.3. Asignatura «Almacenes y minería de datos»

La asignatura «Almacenes y minería de datos» (ver cuadro 3) tiene como objetivo principal adquirir los conocimientos y habilidades necesarios para diseñar e implantar almacenes de datos, así como para aplicar técnicas básicas de minería de datos que permitan extraer conocimiento a partir de los datos. El almacenamiento y la explotación de información digital constituyen un aspecto clave para el éxito de cualquier empresa u organización. Los almacenes de datos pueden considerarse una forma especializada de bases de datos orientadas al análisis de los datos, en lugar de al procesamiento de transacciones, y constituyen el núcleo de los sistemas de ayuda a la toma de decisiones.

En mayor detalle, la asignatura se centra en el diseño de almacenes de datos y en su explotación para extraer información útil a partir de los datos. El primer bloque de la asignatura se centra en los almacenes de datos. En primer lugar se motiva la importancia de los almacenes de datos y la diferencia existente con las bases de datos tradicionales —*Online Analytical Processing* (OLAP) frente a *Online Transaction Processing* (OLTP)—, las distintas arquitecturas y metodologías existentes (incidiendo en las aproximaciones de Ralph Kimball y William H. Inmon) y las aproximaciones tecnológicas disponibles —*Multidimensional OLAP* (MOLAP), *Relational OLAP* (ROLAP) y *Hybrid OLAP* (HOLAP)—. A continuación se aborda el problema de diseño de almacenes de datos utilizando el modelo en estrella. El segundo bloque de la asignatura

Asignatura: Bases de Datos
Materia: Materia Común de Ingeniería de Software y Sistemas de Información
Tecnología Específica: Ninguna (Formación común Rama Informática)
Titulación: Graduado o Graduada en Ingeniería Informática
Universidad: Universidad de Zaragoza
Curso: 2 Cuatrimestre: 4
Carácter: Obligatoria
Departamento: Informática e Ingeniería de Sistemas
Créditos ECTS: 6
Idioma en que se imparte: Castellano
Grupos de teoría: 2
Grupos de problemas: 2
Grupos prácticos de laboratorio: 6

Cuadro 1: Información general sobre la asignatura «Bases de datos»

Asignatura: Bases de Datos 2
Materia: Gestión de Datos e Información
Tecnología Específica: Sistemas de Información, Tecnologías de la Información
Titulación: Graduado o Graduada en Ingeniería Informática
Universidad: Universidad de Zaragoza
Curso: 3 Cuatrimestre: 6
Carácter: Obligatoria en las tecnologías específicas indicadas
Departamento: Informática e Ingeniería de Sistemas
Créditos ECTS: 6
Idioma en que se imparte: Castellano
Grupos de teoría: 1
Grupos de problemas: 1
Grupos prácticos de laboratorio: 1-2

Cuadro 2: Información general sobre la asignatura «Bases de datos 2»

Asignatura: Almacenes y Minería de Datos
Materia: Ayuda a la Toma de Decisiones
Tecnología Específica: Sistemas de Información
Titulación: Graduado o Graduada en Ingeniería Informática
Universidad: Universidad de Zaragoza
Curso: 3 Cuatrimestre: 7
Carácter: Obligatoria en la tecnología específica indicada
Departamento: Informática e Ingeniería de Sistemas
Créditos ECTS: 6
Idioma en que se imparte: Castellano
Grupos de teoría: 1
Grupos de problemas: 1
Grupos prácticos de laboratorio: 1

Cuadro 3: Información general sobre la asignatura «Almacenes y minería de datos»

aborda técnicas de minería de datos cubriendo diversas tareas típicas, entre las que se pueden destacar técnicas de clasificación, técnicas de agrupación (*clustering*) y descubrimiento de reglas de asociación.

Las asignaturas previas «Bases de datos» y «Bases de datos 2» son complementarias, ya que se centran en el diseño de bases de datos tradicionales (adecuadas para el procesamiento de transacciones), pero no en el diseño de almacenes de datos (adecuados para el procesamiento analítico).

B. Encuesta sobre DBDAp – Curso 2015/2016

A continuación se muestra la encuesta realizada el curso 2015/2016 entre el alumnado de las asignaturas «Bases de datos 2» y «Manipulación y análisis de grandes volúmenes de datos».

1. ¿Has utilizado la aplicación DBDAp de forma habitual en la asignatura?
 - Sí
 - No
2. ¿Estás globalmente satisfecho/a con la aplicación DBDAp?
 - Sí
 - No
3. Valora de forma global, del 1 al 10, la aplicación DBDAp
4. Valora, del 1 al 10, las funcionalidades ofrecidas por la aplicación DBDAp
5. Valora, del 1 al 10, la facilidad de uso de la aplicación DBDAp
6. ¿Consideras que resulta de interés seguir mejorando y ampliando la aplicación DBDAp?
 - Sí
 - No
7. Selecciona todas las funcionalidades que hayas utilizado de DBDAp
 - Diseño conceptual utilizando esquemas E/R
 - Diseño conceptual utilizando diagramas de clase UML
 - Traducción de esquemas E/R en esquemas relacionales
 - Traducción de esquemas E/R en esquemas relacionales utilizando el modo de transformación paso a paso

- Definición de notaciones alternativas para el modelo E/R
- Normalización de esquemas relacionales
- Detección automática de dependencias funcionales y/o multivaluadas a partir de un conjunto de datos (*dataset*) dado
- Generación automática de código SQL
- Diseño de almacenes de datos

8. Indica otras aplicaciones similares a DBDAp, o parcialmente relacionadas, que conozcas (si la relación es parcial, indica cuál es)
9. ¿Conoces alguna aplicación de apoyo al aprendizaje y/o diseño de BD mejor que DBDAp o que la supere en alguna funcionalidad? Si es así, indica cuáles y en qué superan a DBDAp
10. Indica las sugerencias de mejora y ampliaciones que te gustaría que se incorporasen a DBDAp
11. En caso de que hayas detectado *bugs* en la herramienta, indícalos aquí
12. Si tienes comentarios o sugerencias adicionales, indícalos aquí

C. Encuesta sobre DBDAp – Curso 2016/2017

A continuación se muestra la encuesta realizada a comienzos del curso 2016/2017 entre el alumnado de la asignatura «Almacenes y minería de datos». Dado que se esperaba que gran parte del alumnado de esta asignatura había cursado ya «Bases de datos 2» el curso pasado, y por tanto probablemente rellenó la encuesta mostrada en el Anexo B, se cambió el contenido de la encuesta y se introdujeron preguntas con escala de Likert, con objeto de ver el grado de conformidad del alumnado con respecto a una serie de afirmaciones relativas a la herramienta DBDAp.

1. ¿Conocías ya la herramienta DBDAp?
 - Sí
 - No
2. ¿Diste tu opinión sobre DBDAp el curso pasado?
 - Sí
 - No

Indica tu grado de conformidad (de 1 –disconformidad total– a 10 –conformidad total–) con las afirmaciones que se indican a continuación

3. Utilizo DBDAp habitualmente
4. DBDAp es una herramienta de interés en asignaturas relacionadas con bases de datos

5. DBDap me ayuda en el aprendizaje de diversos aspectos relacionados con asignaturas de bases de datos
6. DBDap es una buena herramienta como soporte para el diseño de bases de datos y la realización de ejercicios
7. DBDap es fácil de manejar
8. DBDap es potente en cuanto a las funcionalidades ofrecidas
9. Las funcionalidades ofrecidas por DBDap son interesantes en el contexto de la asignatura
10. DBDap facilita el diseño de bases de datos y almacenes de datos
11. Es interesante seguir manteniendo DBDap
12. Es interesante seguir ampliando DBDap
13. No existe otra herramienta disponible con funcionalidades similares para el aprendizaje y docencia de asignaturas relacionadas con bases de datos
14. Indica qué aspectos te permite aprender mejor la herramienta
15. Si tienes comentarios o sugerencias adicionales, indícalos aquí

Referencias

- [1] Christopher Adamson. *Star Schema: The Complete Reference*. McGraw-Hill, 2010.
- [2] Gary Anthes. Happy birthday, RDBMS! *Communications of the ACM*, 53(5):16–17, 2010.
- [3] Sikha Bagui. *Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends*, chapter 1 “Mapping Generalizations and Specializations and Categories to Relational Databases”, pages 1–11. IGI Publishing, 2009.
- [4] Catriel Beeri y Philip A. Bernstein. Computational problems related to the design of normal form relational schemas. *ACM Transactions on Database Systems*, 4(1):30–59, 1979.
- [5] P. A. Bernstein, J. R. Swenson y D. C. Tsichritzis. A unified approach to functional dependencies and relations. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 237–245. ACM, 1975.
- [6] Philip A. Bernstein. Synthesizing third normal form relations from functional dependencies. *ACM Transactions on Database Systems*, 1(4):277–298, 1976.
- [7] Miloš Bogdanović, Aleksandar Stanimirović, Nikola Davidović y Leonid Stoimenov. The development and usage of a relational database design tool for educational purposes. In *Informing Science & IT Education Conference (InSITE)*, pages 251–258, 2008. ISSN 1535-0703.
- [8] Peter Brusilovsky, Sergey Sosnovsky, Michael V. Yudelton, Danielle H. Lee, Vladimir Zadorozhny y Xin Zhou. Learning SQL programming with interactive tools: From integration to personalization. *ACM Transactions on Computing Education*, 9(4):19:1–19:15, 2010.
- [9] Coral Calero, Mario Piattini y Francisco Ruiz. Towards a database body of knowledge: A study from Spain. *SIGMOD Record*, 32(2):48–53, 2003.
- [10] Rick Cattell. Scalable SQL and NoSQL data stores. *SIGMOD Record*, 39(4):12–27, 2010.
- [11] Peter Pin-Shan Chen. The entity-relationship model — Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.
- [12] Roberta Cochrane, Hamid Pirahesh y Nelson Mendonça Mattos. Integrating triggers and declarative constraints in SQL database systems. In *22th International Conference on Very Large Data Bases (VLDB)*, pages 567–578. Morgan Kaufmann Publishers Inc., 1996.
- [13] E. F. Codd. Recent investigations in relational data base systems. In *IFIP Congress*, pages 1017–1021, 1974.
- [14] Edgar Frank Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.
- [15] Thomas M. Connolly y Carolyn E. Begg. A constructivist-based approach to teaching database analysis and design. *Journal of Information Systems Education*, 17(1):43–53, 2006.
- [16] Dolores Cuadra, Ana Iglesias, Elena Castro y Paloma Martínez. Educational experiences detecting, using, and representing ternary relationships in database design. *IEEE Transactions on Education*, 53(3):358–364, 2010.
- [17] Adoración de Miguel, Mario Piattini y Esperanza Marcos. *Diseño de Bases de Datos Relacionales*. Alfaomega, 2000.
- [18] Benjamin S. Bloom (editor). *Taxonomy of Educational Objectives, Handbook 1: Cognitive Domain*. Longman, 1956.
- [19] Andrew Eisenberg y Jim Melton. SQL: 1999, formerly known as SQL3. *SIGMOD Record*, 28(1):131–138, 1999.
- [20] Ramez Elmasri y Shamkant Navathe. *Fundamentals of Database Systems*. Addison-Wesley Publishing Company, USA, sixth edition, 2010.

- [21] Ronald Fagin. Multivalued dependencies and a new normal form for relational databases. *ACM Transactions on Database Systems*, 2(3):262–278, 1977.
- [22] Vahid Garousi. Applying peer reviews in Software Engineering education: An experiment and lessons learned. *IEEE Transactions on Education*, 53(2):182–193, 2010.
- [23] Jim Gray, Surajit Chaudhuri, Adam Bosworth, Andrew Layman, Don Reichart, Murali Venkatrao, Frank Pellow y Hamid Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Mining and Knowledge Discovery*, 1(1):29–53, 1997.
- [24] Jeffrey A. Hoffer, Ramesh Venkataraman y Heikki Topi. *Modern Database Management*. Prentice Hall, tenth edition, 2010.
- [25] Sergio Ilarri y Sergio González. Herramienta de apoyo a la docencia en el diseño de bases de datos y almacenes de datos. In *XX Jornadas de Ingeniería del Software y Bases de Datos (JISBD), Demo*, 2015.
- [26] William Kent. A simple guide to five normal forms in relational database theory. *Communications of the ACM*, 26(2):120–125, 1983.
- [27] Nurul Naslia Khairuddin y Khairuddin Hashim. Application of Bloom’s taxonomy in software engineering assessments. In *Eighth WSEAS International Conference on Applied Computer Science (ACS)*, pages 66–69. World Scientific and Engineering Academy and Society (WSEAS), 2008.
- [28] Ralph Kimball. Dimensional modeling techniques. Kimball Group, <http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/> [Accessed October 28, 2016].
- [29] Ralph Kimball y Joe Caserta. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming and Delivering Data*. John Wiley & Sons, 2004.
- [30] Ralph Kimball y Margy Ross. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley Publishing, third edition, 2013.
- [31] Ilana Lavy y Aharon Yadin. Team-based peer review as a form of formative assessment – the case of a systems analysis and design workshop. *Journal of Information Systems Education*, 21(1):85–98, 2010.
- [32] Neal Leavitt. Will NoSQL databases live up to their promise? *Computer*, 43(2):12–14, 2010.
- [33] María Mercedes Martínez-González y Gérald Duffing. Teaching databases in compliance with the European dimension of Higher Education: Best practices for better competences. *Education and Information Technologies*, 12(4):211–228, 2007.
- [34] John A. Mendonca. Implementing peer technical reviews in a large-sized database course. In *Information Systems Education Conference*, 2000.
- [35] Antonija Mitrovic. An intelligent SQL tutor on the Web. *International Journal of Artificial Intelligence in Education*, 13:171–195, 2003.
- [36] Meg Murray y Mario Guimaraes. Animated database courseware: using animations to extend conceptual understanding of database concepts. *Journal of Computing Sciences in Colleges*, 24(2):144–150, 2008.
- [37] Meg Murray y Mario Guimaraes. Expanding the database curriculum. *Journal of Computing Sciences in Colleges*, 23(3):69–75, 2008.
- [38] Meg Murray y Mario Guimaraes. Animated courseware support for teaching database design. *Issues in Informing Science and Information Technology*, 6:201–211, 2009.
- [39] Bruce J. Neubauer. Data modeling in the understanding database course: Adding UML and XML modeling to the traditional course content. *Journal of Computing Sciences in Colleges*, 17(5):147–153, 2002.
- [40] Thorsten Papenbrock, Jens Ehrlich, Jannik Marten, Tommy Neubert, Jan-Peer Rudolph, Martin Schönberg, Jakob Zwiener y Felix Naumann. Functional dependency discovery: An experimental evaluation of seven algorithms. *Proceedings of the VLDB Endowment*, 8(10):1082–1093, 2015.
- [41] Luisa M. Regueras, Elena Verdú, María J. Verdú, María Á. Pérez y Juan P. de Castro. E-learning strategies to support databases courses: A case study. In *International Conference on Technology, Training and Communication*, volume 361. CEUR Workshop Proceedings, 2007. 8 pages.
- [42] James Rumbaugh. ER is UML. *Journal of Information Systems Education*, 17(1):21–26, 2006.
- [43] James Rumbaugh, Ivar Jacobson y Grady Booch. *El Lenguaje Unificado de Modelado: Manual de Referencia*. Pearson Educación, 2007.
- [44] Devang Shah y Sandra Slaughter. *UML and the Unified Process*, chapter X “Transforming UML Class Diagrams Into Relational Data Models”, pages 217–236. IGI Publishing, 2003.

- [45] Guttorm Sindre, Daniel L. Moody, Terje Brasethvik y Arne Solvberg. Introducing peer review in an IS analysis course. *Journal of Information Systems Education*, 14(1):101–119, 2003.
- [46] Henry C. Smith. Database design: Composing fully normalized tables from a rigorous dependency diagram. *Communications of the ACM*, 28(8):826–838, 1985.
- [47] Michael Stonebraker. SQL databases v. NoSQL databases. *Communications of the ACM*, 53(4):10–11, 2010.
- [48] Michael Stonebraker. Stonebraker on NoSQL and enterprises. *Communications of the ACM*, 54(8):10–11, 2011.
- [49] James Suleiman y Monica J. Garfield. Conceptual data modeling in the introductory database course: Is it time for UML? *Journal of Information Systems Education*, 17(1):93–100, 2006.
- [50] Pramuditha Suraweera y Antonija Mitrovic. KERMIT: A constraint-based tutor for database modeling. In *Intelligent Tutoring Systems*, volume 2363 of *Lecture Notes in Computer Science (LNCS)*, pages 377–387. Springer, 2002.
- [51] Toby J. Teorey, Dongqing Yang y James P. Fry. A logical design methodology for relational databases using the extended entity-relationship model. *ACM Computing Surveys*, 18(2):197–222, 1986.
- [52] Keith Topping. Peer assessment between students in colleges and universities. *Review of Educational Research*, 68(3):249–276, 1998.
- [53] Miguel Valero-García y Luis M. Díaz de Cerio. Autoevaluación y co-evaluación: estrategias para facilitar la evaluación continuada. In *Simposio Nacional de Docencia en la Informática (SINDI)*, pages 25–32. Thomson, 2005.
- [54] Keith Willey y Mark Freeman. Completing the learning cycle: The role of formative feedback when using self and peer assessment to improve teamwork and engagement. In *17th Annual Conference of the Australasian Association for Engineering Education: Creativity, Challenge, Change; Partnerships in Engineering Education*, pages 751–758. Australasian Association for Engineering Education, 2006.



Sergio Ilarri Artigas es Ingeniero en Informática por la Universidad de Zaragoza y doctor por el programa de doctorado en Ingeniería de Sistemas e Informática de dicha universidad. Es profesor del Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza. Imparte docencia fundamentalmente en materias relacionadas con bases de datos e ingeniería del software. Sus áreas de investigación incluyen la gestión de datos para computación móvil y sistemas distribuidos, las redes de vehículos, y la Web Semántica.



2017 S. Ilarri. Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional que permite copiar, distribuir y comunicar públicamente la obra en cualquier medio, sólido o electrónico, siempre que se acrediten a los autores y fuentes originales y no se haga un uso comercial.