



UNIVERSIDAD
COMPLUTENSE
MADRID

Proyecto de Innovación

Convocatoria 2017/2018

Nº de proyecto 114

Título del proyecto

Mejora del aprendizaje de SQL con realimentación semántica

Nombre del responsable del proyecto

Fernando Sáenz Pérez

Centro

Facultad de Informática

Departamento

Departamento de Ingeniería del Software e Inteligencia Artificial

1. Objetivos propuestos en la presentación del proyecto

Bajo la directiva de mejora del aprendizaje de SQL con realimentación semántica, se identificaron los siguientes objetivos de este proyecto:

1- Mejora del análisis de consultas SQL en DES aplicando técnicas semánticas con objeto de informar al alumno de que, aún escribiendo una consulta sintácticamente correcta, pueda ser incorrecta desde el punto de vista de su resultado al ejecutarla (es decir, que pueda ser semánticamente incorrecta). Una primera alternativa es el análisis de las consultas con respecto a su forma y los metadatos. Esto incluye la verificación de fórmulas lógicas que pueden reducirse en última instancia a trivialmente ciertas o falsas. De este modo se pueden identificar respectivamente simplificaciones e incorrecciones. Una segunda alternativa consiste en tener en cuenta también la instancia de la base de datos para determinar errores semánticos dependientes del contexto. Como inconveniente en esta segunda alternativa se puede destacar que el tiempo de análisis requerido aumentará en general dado que habrá que resolver consultas concretas en el sistema gestor para analizar sus resultados. Además, al tratarse de un enfoque contextual, los resultados del análisis dependerían de la instancia concreta sobre la que se ha efectuado, pudiendo variar las conclusiones en otras instancias. No obstante, para ciertas bases de datos suficientemente estables, puede tratarse de una alternativa útil. Finalmente, al razonar sobre fórmulas lógicas es posible determinar formulaciones equivalentes simplificadas. Esto permitiría plantear automáticamente al estudiante consultas alternativas mejores desde el punto de vista de la concreción.

2- Aumento de la visibilidad del sistema DES para la comunidad educativa internacional actualizando su página web y aportándole imagen de identidad (corporativa). Personal externo al equipo que presenta esta solicitud y especializado en diseño de páginas web e identidad corporativa sería el responsable de aportar una página web acorde con la estética y funcionalidades actuales, además de proporcionar logos, iconos y material gráfico tanto a la página web como a la herramienta.

3- Facilitar el acceso a la herramienta mediante una interfaz de usuario web atractiva que proporcione las siguientes ventajas. Por una parte, formadores y profesores pueden obtener unas primeras impresiones sobre la herramienta al evaluarla para su adopción en clase sin invertir el tiempo de su descarga e instalación. Por otra parte, facilitar su uso al proporcionar un acceso inmediato y sin necesidad de configuración en entornos en donde no se permita o se condicione la instalación de software de terceros. Esta interfaz irá acompañada de ejemplos y casos de uso para motivar su uso y facilitar las primeras tomas de contacto.

4- Evaluación docente de los resultados de la herramienta. Se solicitará la realimentación de los estudiantes al realizar las prácticas regladas de las asignaturas de bases de datos para obtener, por una parte, casos de uso objetivos en los que la herramienta ha sido de utilidad y, por otra, conseguir una valoración subjetiva. Para el análisis objetivo se recabarán los registros (log) del sistema para automáticamente detectar esos casos, y para el análisis subjetivo se proporcionará una breve encuesta de satisfacción acerca de la herramienta.

2. Objetivos alcanzados

Los objetivos alcanzados han sido los siguientes (identificados según la numeración de la propuesta original):

1- Mejora del análisis de consultas SQL en DES aplicando técnicas semánticas.

Se ha conseguido incorporar en el sistema DES el análisis semántico de consultas SQL para el aprendizaje de este lenguaje. En particular se detectan los siguientes errores, que están numerados según se describe en el artículo [Brass&Goldberg'06] "Stefan Brass & Christian Goldberg (2006): Semantic Errors in SQL Queries: A Quite Complete List. The Journal of Systems and Software 79(5)":

Error 1: Condición inconsistente.

Error 2: Cláusula DISTINCT innecesaria.

Error 3: Columna de salida constante.

Error 4: Valores de columnas duplicados en la salida.

Error 5: Variable de tupla (referencia a relación) sin uso.

Error 6: Reunión (join) innecesaria.

Error 7: Variables de tupla idénticas (dos o más relaciones producen las mismas tuplas).

Error 8: Condición implicada o tautológica.

Error 9: Comparación con NULL.

Error 11: Operador de comparación innecesariamente general.

Error 12: Patrón LIKE sin comodines.

Error 13: Sentencia SELECT innecesariamente complicada en una subconsulta EXISTS.

Error 16: Cláusula DISTINCT innecesaria en función de agregación.

Error 17: Argumento innecesario en COUNT.

Error 27: Condición de reunión (join) ausente.

Error 32: Cláusula HAVING sin agrupación (GROUP BY).

Error 33: Usos posiblemente inadecuados de DISTINCT en funciones de agregación.

3- Facilitar el acceso a la herramienta mediante una interfaz de usuario web

Este objetivo ha podido conseguirse aunque su implementación corría a cargo del personal externo no concedido. Dado que el objetivo 4 (no alcanzado como se describe más adelante) no se ha realizado, se han usado los recursos humanos destinados a él en este otro objetivo. De este modo, esperamos un mayor impacto de la herramienta en la comunidad educativa porque evita los posibles problemas de su instalación. De hecho, la realimentación recibida por los canales de comunicación del sistema DES apuntaban a esta interfaz como un objetivo de interés, de forma parecida a herramientas como SWISH (interfaz web para SWI-Prolog), que facilitan el contacto con los sistemas mediante navegadores web.

Además, dentro del plan de trabajo se incluyó el apartado de difusión (punto 6) en el que se proponía la publicación en congresos o revistas de los resultados del trabajo. Se envió la comunicación "Semantic

Analysis of SQL Statements in DES" al congreso PROLE de la sociedad SISTEDES, que fue aceptada para su presentación en el próximo mes de septiembre. Sus contenidos se centran en la descripción de la herramienta con todos los tipos de errores semánticos identificados. Por otro lado, se ha redactado y enviado para su revisión un artículo de revista: "Constraint Logic Programming for SQL Semantic Checking" que hace énfasis en las técnicas usadas para la detección de inconsistencias, tautologías y simplificaciones mediante programación lógica con restricciones y cooperación de resolutores. Finalmente, también se ha publicado la nueva versión de DES resultado de este proyecto en las listas de distribución de SWI-Prolog, SICStus Prolog, RedBD y Datalog Cafe.

Los objetivos no alcanzados han sido los siguientes:

2- Aumento de la visibilidad del sistema DES

Esta actividad estaba asignada al personal externo para el que no obtuvimos financiación. Por lo tanto, la página web del sistema DES no ha experimentado ningún cambio significativo salvo la actualización debida a la nueva distribución resultado de este proyecto. En particular se ha añadido la nueva página de descargas para distintos sistemas operativos (Windows 10, Mac OS X High Sierra y Ubuntu 18.04 LTS), la página de novedades (Release Notes) y descripción (What's New) de la versión, y el manual on-line.

4- Evaluación docente de los resultados de la herramienta

Dado que las asignaturas sujeto de estudio estaban planificadas en el primer cuatrimestre, no ha sido posible la implantación de la herramienta en el aula. No obstante, sí se ha hecho pública la herramienta (DES versión 6.0) el día 9 de abril de 2018 y desde entonces se han contabilizado 367 descargas (a día 4 de junio de 2018). Menos de dos meses más tarde se ha publicado una nueva versión con mejoras en particular del análisis semántico (DES versión 6.1 a finales de mayo). La evaluación docente de la herramienta es objetivo pendiente y objeto del proyecto de innovación docente que se ha solicitado y concedido para el próximo curso 2018-19, denominado "Evaluación de la realimentación semántica de SQL en DES". El nuevo equipo de trabajo tendrá una primera reunión en este mismo mes de junio.

3. Metodología empleada en el proyecto

Se ha empleado una metodología clásica de desarrollo de software con las etapas de diseño, implementación, pruebas, ajuste y paso a producción. En particular:

1- Diseño. Se han identificado los casos que presentan síntomas de errores según las referencias [Brass&Goldberg'06, Guo et al'96] (véase el anexo). También se ha evaluado la complejidad de diseño e implementación de los casos con las técnicas de programación lógica con restricciones, la cooperación de dominios de restricciones, y el examen del árbol sintáctico de las consultas SQL. Se han diseñado los algoritmos para la identificación de los casos seleccionados: errores 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 16, 17, 27, 32, 33 según se describen en el apartado anterior. Se ha diseñado el gestor de avisos semánticos que opera en dos etapas: primero en la etapa de análisis sintáctico de las sentencias, y posteriormente en la etapa de compilación. El resultado de este gestor es un informe de los errores semánticos detectados en la sentencia. Con respecto a la interfaz de usuario web al sistema, se ha diseñado una interfaz que incluye un editor, una vista de archivos de usuario en la nube, un panel de salida de mensajes, y un panel interactivo para el uso del sistema.

2. Implementación. Se han implementado los algoritmos definidos en la etapa de diseño en un nuevo módulo Prolog (`des_sql_semantic.pl`). Este módulo hace uso de los resolutores de restricciones nativos CLP(R), CLP(Q) y CLP(FD), además de las restricciones nativas del resolutor de Herbrand (igualdad y desigualdad sintácticas). El gestor de avisos se ha implementado en conexión con el módulo de análisis y compilación de SQL (`des_sql.pl`). La interfaz web se ha implementado con páginas HTML 5 dinámicas generadas por SWI-Prolog bajo demanda de los navegadores clientes. Además se ha incorporado JavaScript para el procesamiento en el lado del cliente, como el coloreado sintáctico y los elementos activos (cronómetro, botones, etc.). El estilo se define con hojas CSS. Por su parte, el servidor web se ha implementado en conexión con procesos del sistema que ejecutan instancias distintas de DES para los distintos usuarios. El sistema cuenta con rol de administración para gestión de usuarios y recursos, y rol plano para usuarios externos (acceso anónimo).

3. Pruebas. Se ha seleccionado un conjunto de casos de prueba en un rango razonable de ejemplos de uso (consultas y vistas) y generado de una familia de tests (test suite) para automatizar la validación. Se ha comprobado y ajustado la interfaz web al sistema en comunicación con los servicios web, haciendo especial hincapié en la protección de recursos: tiempo de ejecución y memoria en disco.

4. Paso a producción. Se han publicado las versiones fuentes y ejecutables del sistema para su uso local mediante descarga e instalación. Actualmente se está instalando el nuevo sistema operativo Ubuntu 18.04 LTS en una de las máquinas servidoras del grupo (dalila.sip.ucm.es) para alojar el sistema web interactivo. Para ello se contará con la instalación de un certificado digital que permita el acceso seguro mediante https, de modo que las conexiones sean cifradas y no se transmita ningún dato plano. En abril de este año se publicó la página web de la primera versión (6.0) en des.sourceforge.net y su mejora (6.1) en mayo.

4. Recursos humanos

En este proyecto han intervenido:

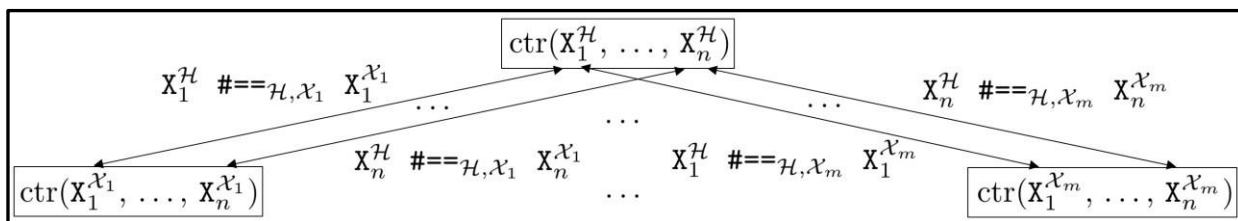
- Rafael Caballero Roldán, del Departamento Sistemas Informáticos y Computación -Lenguajes y Sistemas Informáticos y Ciencias de la Computación e Inteligencia Artificial- de la Facultad de Informática.
- Yolanda García Ruíz, del Departamento Sistemas Informáticos y Computación -Lenguajes y Sistemas Informáticos y Ciencias de la Computación e Inteligencia Artificial- de la Facultad de Informática.
- Luis Garmendia Salvador, del Departamento Ingeniería de Software e Inteligencia Artificial - Lenguajes y Sistemas Informáticos- de la Facultad de Informática.
- Fernando Sáenz Pérez, del Departamento Ingeniería de Software e Inteligencia Artificial - Lenguajes y Sistemas Informáticos- de la Facultad de Informática.

Se solicitó financiación para la contratación de personal externo con objeto de participar en dos actividades: el desarrollo de la página web para difusión de resultados, y la interfaz web al sistema para el acceso on-line desde navegadores web. Al no recibir esta financiación, no hemos contado con su ayuda en estas actividades.

5. Desarrollo de las actividades

1. Análisis de sentencias SQL

Hemos propuesto y desarrollado el uso de la programación lógica con restricciones como marco de razonamiento sobre propiedades de las sentencias SQL independientes de cualquier estado de la base de datos (instancia). Para ello, traducimos una sentencia SQL en un programa lógico con restricciones que en particular modela las condiciones y las expresiones en las listas de selección SQL. En la traducción, una variable representa los posibles valores que un puede tomar una columna de una variable de relación para cada instancia en el contexto de todas las relaciones involucradas en la sentencia SQL. La traducción incluye técnicas de optimización y simplificación como el plegado y el desplegado. Este programa CLP traducido se evalúa para obtener las propiedades de interés para el análisis semántico. Como ejemplos de identificación de estas propiedades mencionamos los siguientes: un fallo en la deducción indica una condición inconsistente; si se determinan vinculaciones cerradas, significa que ciertas propiedades son ciertas independientemente de la instancia de la base de datos; finalmente, si la deducción de la traducción de una condición complementada falla, significa que la condición es una tautología. El uso de resolutores de restricciones permite la deducción lógica involucrando condiciones con variables sin instanciar. Sin embargo, cada uno de los resolutores de restricciones disponibles en los sistemas de programación lógica presentan distintos grados de deducción debido a sus mecanismos de propagación y etiquetado. Por ello, aplicamos una técnica genérica de cooperación de resolutores que se muestra en la siguiente figura para aumentar el poder de deducción al colaborar mediante la instanciación de variables en los respectivos dominios:



En nuestro caso permitimos la cooperación de los resolutores numéricos CLP(FD) y CLP(Q) para tipos numéricos enteros (además de la participación individual para los dominios de números fraccionarios y cadenas de caracteres de los resolutores CLP(R) y Herbrand, respectivamente). Nuestro enfoque permite en particular detectar los siguientes casos que mostramos como ejemplo:

```
DES> CREATE TABLE gas_products(
name VARCHAR(20) PRIMARY KEY,
butane FLOAT CHECK butane BETWEEN 0 AND 100,
propane FLOAT CHECK propane BETWEEN 0 AND 100,
olefins FLOAT CHECK olefins BETWEEN 0 AND 100,
diolefins FLOAT CHECK diolefins BETWEEN 0 AND 100,
CHECK butane+propane+olefins+diolefins = 100);
DES> SELECT name FROM gas_products WHERE butane>60 AND propane>50;
Warning: Inconsistent condition.
```

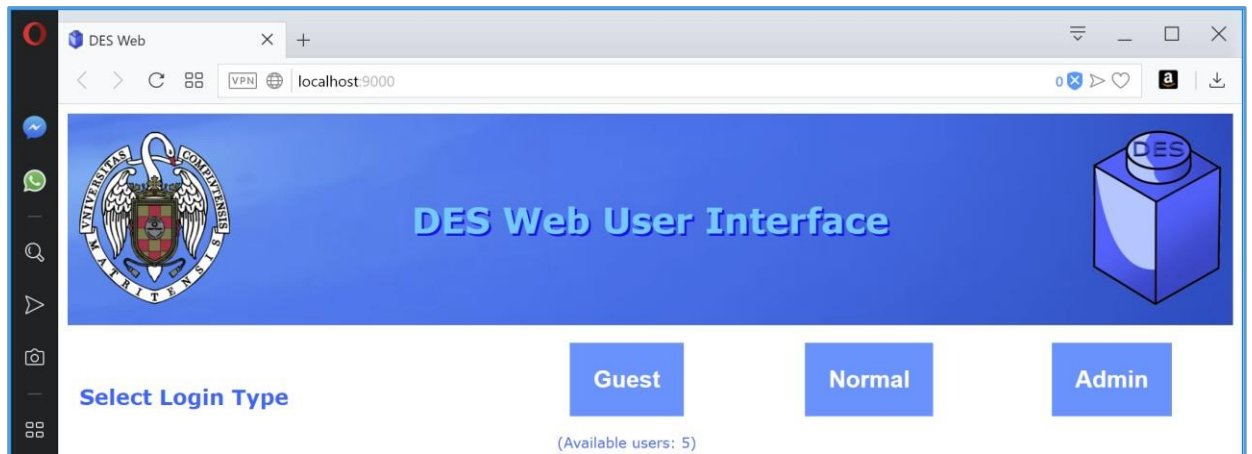
En este primer ejemplo, la definición de las restricciones en la tabla implica que la condición que se impone sobre la sentencia SELECT es insatisfactible independientemente de los datos de la tabla.

```
DES> SELECT butane, propane FROM gas_products WHERE butane-propane=10
AND butane+propane=80;
Warning: Constant output column "butane" with value "45.0".
Warning: Constant output column "propane" with value "35.0".
```

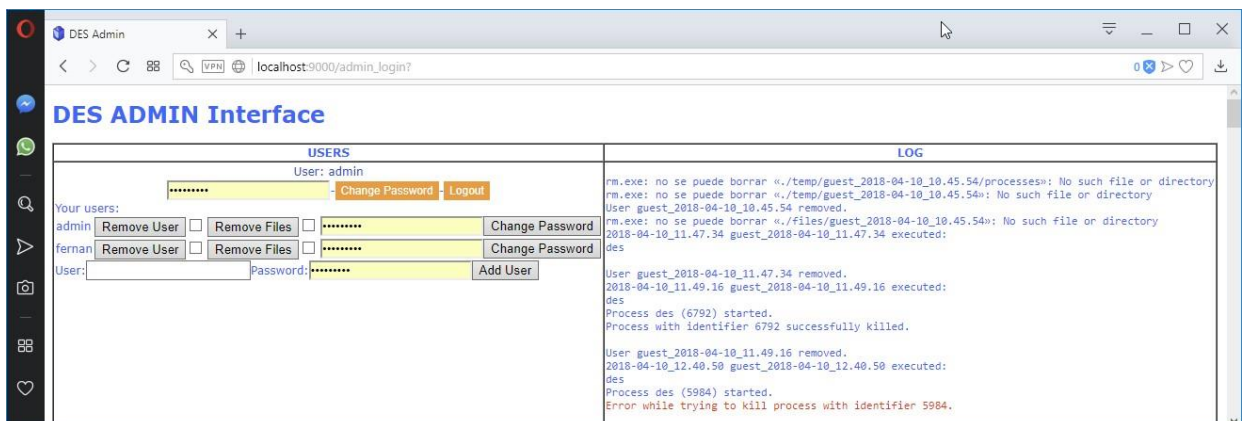
En este segundo ejemplo se determina que la condición impuesta solo se puede cumplir para valores fijos de dos columnas. Se pueden encontrar más ejemplos en <http://www.fdi.ucm.es/profesor/fernan/DES/prole2018/examples.pdf>.

2. Interfaz web al sistema

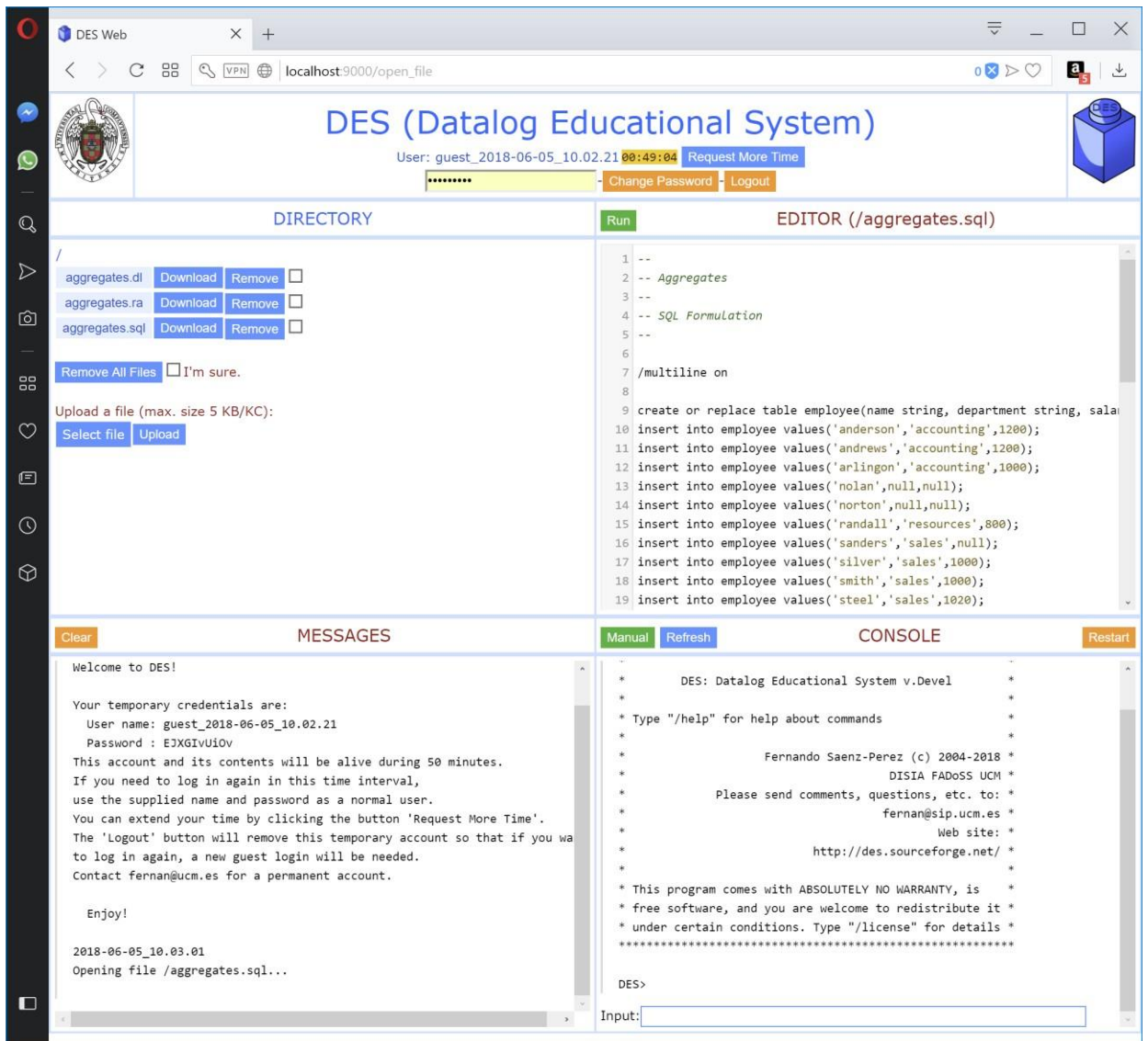
Hemos desarrollado una interfaz web al sistema DES que implementa un IDE (Integrated Development Environment) básico para interactuar con el sistema. La interfaz permite los siguientes tipos de acceso que se muestran en su fachada:



Admin permite la gestión administrativa de usuarios (nombres, contraseñas, borrado, adición, ...) y sus ficheros. En el panel izquierdo que se muestra en la siguiente figura se pueden efectuar estas acciones, mientras que en el derecho se muestran los mensajes resultado de estas acciones. También se recogen los mensajes del sistema en cuanto a accesos, cuotas, timeouts y todos los sucesos críticos desde un punto de vista administrativo.

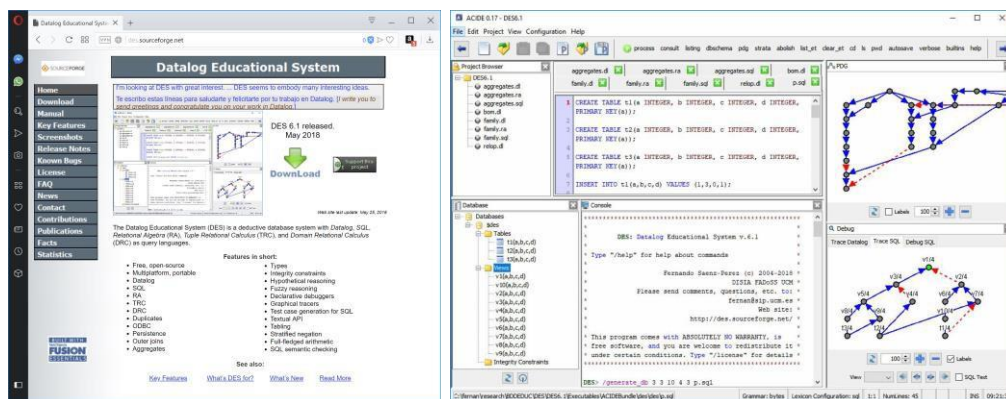


El acceso *Normal* permite el acceso a usuarios registrados, y el acceso *Guest* permite el acceso a usuarios anónimos por un tiempo máximo predeterminado, que puede extenderse a petición del propio usuario. La siguiente figura muestra la interfaz para estos dos tipos de usuario dividida en paneles: DIRECTORY, con el contenido de los ficheros del usuario (con operaciones para cargar, descargar, editar y eliminar ficheros). El panel EDITOR permite la edición de ficheros (con coloreado sintáctico) y su procesamiento en el panel CONSOLE. Este panel muestra la salida del sistema DES y permite en particular la entrada de instrucciones SQL. También permite el acceso al manual del sistema. El último panel, MESSAGES, muestra los mensajes de salida de la interfaz interactiva. La cabecera muestra el nombre de usuario, el tiempo de conexión remanente, permitiendo solicitar su extensión con el botón Request More Time. También es posible cambiar la contraseña y cerrar la sesión. Es importante resaltar que esta herramienta no exige registro alguno para poder usarla. La asignación de un usuario y contraseñas arbitrarios a cada usuario *Guest* se proporciona para reanudar sesión en otro momento conservado el estado de la cuenta (con todos sus ficheros). El administrador controla la cuota de los usuarios y el sistema vigila que no se sobrepasen tanto esta cuota como el tiempo máximo de sesión.



3. Página web del sistema

En des.sourceforge.net se puede ver la página web del sistema, que incluye descargas para las distintas versiones de la herramienta DES: programas de escritorio para terminales (Windows, Ubuntu y Mac OS X), aplicaciones Windows, y un IDE gráfico programado en Java con más funcionalidades que las de la interfaz web. La siguiente es una captura de esta página web y, a su lado, del IDE Java:



6. Anexos

Bibliografía referenciada en esta memoria

[Brass&Goldberg'06]Stefan Brass & Christian Goldberg (2006): *Semantic Errors in SQL Queries: A Quite Complete List*. The Journal of Systems and Software 79(5).

[Guo *et al'*96] Sha Guo, Wei Sun, and Mark A. Weiss. *Solving satisfiability and implication problems in database systems*. ACM Transactions on Database Systems 21, 270–293, 1996.

Bibliografía principal usada en el proyecto

Abello, A., Burgues, X., Casany, M., Martin, C., Quer, C., Rodriguez, M. E., Romero, O., & Urpi, T. (2016). A software tool for e-assessment of relational database skills. *International journal of engineering education*, 32 (3A), 1289-1312. URL: <http://hdl.handle.net/2117/89668>; http://www.ijee.ie/latestissues/Vol32-3A/23_ijee3260ns.pdf.

Albert, E., Gómez-Zamalloa, M., & Puebla, G. (2009). Test Data Generation of Bytecode by CLP Partial Evaluation. In 18th International Symposium on Logic-based Program Synthesis and Transformation (LOPSTR'08) number 5438 in Lecture Notes in Computer Science (pp. 4-23). Springer-Verlag.

Allen, G. N. (2000). WebSQL: An Interactive Web Tool for Teaching Structured Query Language. In AMCIS 2000 Proceedings 384.

Apt, K. (2003). *Principles of Constraint Programming*. New York, NY, USA: Cambridge University Press.

Brass, S., & Goldberg, C. (2006). Semantic Errors in SQL Queries: A Quite Complete List. *The Journal of Systems and Software*, 79 (5), 630-644. URL: <http://dx.doi.org/10.1016/j.jss.2005.06.028>. doi:10.1016/j.jss.2005.06.028.

Caballero, R., & Ieva, C. (2015). Constraint Programming Meets SQL . In XV Jornadas sobre Programación y Lenguajes, PROLE 2015 (SISTEDES).

Correas, J., Martín, S. E., & Sáenz-Pérez, F. (2018). Enhancing set constraint solvers with bound consistency. *Expert Systems with Applications*, 92 , 485 - 494. URL: <http://www.sciencedirect.com/science/article/pii/S0957417417306620>. doi:<https://doi.org/10.1016/j.eswa.2017.09.056>.

Danaparamita, J., & Gatterbauer, W. (2011). Queryviz: Helping users understand sql queries and their patterns. In *Proceedings of the 14th International Conference on Extending Database Technology EDBT/ICDT'11* (pp. 558-561). New York, NY, USA: ACM. URL: <http://doi.acm.org/10.1145/1951365.1951440>. doi:10.1145/1951365.1951440.

Frühwirth, T. (2009). *Constraint Handling Rules*. (1st ed.). New York, NY, USA: Cambridge University Press.

Guagliardo, P., & Libkin, L. (2017). A Formal Semantics of SQL Queries, Its Validation, and Applications. *Proceedings of the VLDB Endowment*, 11 (1), 27-39. URL: <https://doi.org/10.14778/3151113>. doi:10.14778/3151113.3151116.

Hitzler, P., & Seda, A. K. (2011). *Mathematical Aspects of Logic Programming Semantics*. Chapman and Hall / CRC studies in informatics series. CRC Press.

Hofstedt, P. (2000). Cooperating Constraint Solvers. In R. Dechter (Ed.), *Sixth International Conference on Principles and Practice of Constraint Programming - CP*. Springer-Verlag volume 1894 of LNCS.

ISO/IEC (2016). SQL:2016 ISO/IEC 9075-1:2016 Standard.

Jaffar, J., & Lassez, J.-L. (1987). Constraint logic programming. In *Proceedings of the 14th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages POPL '87* (pp. 111-119). New York, NY, USA: ACM. URL: <http://doi.acm.org/10.1145/41625.41635>. doi:10.1145/41625.41635.

Javid, M., Embury, S., Srivastava, D., & Ari, I. (2012). Diagnosing faults in embedded queries in database applications. In D. Srivastava, & I. Ari (Eds.), *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, Berlin, Germany, March 30, 2012 (pp. 239-244). United States: Association for Computing Machinery. doi:10.1145/2320765.2320831.

Jones, N. D. (1996). An introduction to partial evaluation. *ACM Comput. Surv.*, 28 (3), 480-503. URL: <http://doi.acm.org/10.1145/243439.243447>. doi:10.1145/243439.243447.

Leuschel, M., & Bruynooghe, M. (2002). Logic program specialisation through partial deduction: Control issues. *TPLP*, 2 (4-5), 461-515. URL: <https://doi.org/10.1017/S147106840200145X>. doi:10.1017/S147106840200145X.

Martín, S. E., Hortalá-González, M. T., Rodríguez-Artalejo, M., del Vado Vírseda, R., Sáenz-Pérez, F., & Fernández, A. J. (2009). On the cooperation of the constraint domains h, R and FD in CFLP. *CoRR*, abs/0904.2136 . URL: <http://arxiv.org/abs/0904.2136>. arXiv:0904.2136.

Monfroy, E., & Castro, C. (2004). A component language for hybrid solver cooperations. In *ADVIS* (pp. 192-202). Springer volume 3261 of *Lecture Notes in Computer Science*.

Reza, H., & Zarns, K. (2011). Testing Relational Database Using SQLLint. In *2011 Eighth International Conference on Information Technology: New Generations* (pp. 608-613). doi:10.1109/ITNG.2011.208.

Russell, S. J., & Norvig, P. (2010). *Artificial Intelligence - A Modern Approach* (3. internat. ed.). Pearson Education. URL: http://vig.pearsoned.com/store/product/1,1207,store-12521_isbn-0136042597,00.html.

Sadiq, S., Orłowska, M., Sadiq, W., & Lin, J. (2004). Sqlator: An online sql learning workbench. In *Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education ITiCSE '04* (pp. 223-227). New York, NY, USA: ACM. URL: <http://doi.acm.org/10.1145/1007996.1008055>. doi:10.1145/1007996.1008055.

Sáenz-Pérez, F. (2011). DES: A Deductive Database System. *Electronic Notes on Theoretical Computer Science*, 271 , 63-78.

Sáenz-Pérez, F. (2014). Towards bridging the expressiveness gap between relational and deductive databases. *Electronic Communications of the EASST*, 64 , 1-22.

Sáenz-Pérez, F. (2017). Intuitionistic Logic Programming for SQL. In M. V. Hermenegildo, & P. Lopez-Garcia (Eds.), *Logic-Based Program Synthesis and Transformation* (pp. 293-308). Cham: Springer International Publishing.

Scott, J. D., Flener, P., & Pearson, J. (2013). Bounded strings for constraint programming. In *2013 IEEE 25th International Conference on Tools with Artificial Intelligence* (pp. 1036-1043). doi:10.1109/ICTAI.2013.155.

Silberschatz, A., Korth, H., & Sudarshan, S. (2010). Database Systems Concepts. (Sixth ed.). New York, NY, USA: McGraw-Hill, Inc.

Soler, J., Prados, F., Boada, I., & Poch, J. (2006). A Web-based tool for teaching and learning SQL. In International Conference on Information Technology Based Higher Education and Training, ITHET.

Sterling, L., & Shapiro, E. (1994). The Art of Prolog (2Nd Ed.): Advanced Programming Techniques. Cambridge, MA, USA: MIT Press.

Ullman, J. D. (1988). Principles of Database and Knowledge-Base Systems, Volume II . Computer Science Press.