

# LearnSQL: un juez para el aprendizaje de las bases de datos<sup>1</sup>

Jesús Correas, Enrique Martín, Manuel Montenegro, Adrián Riesco y Rubén Rubio  
Facultad de Informática - Universidad Complutense de Madrid

## Introducción

Los jueces automáticos son aplicaciones web que permiten a sus usuarios enviar una solución a un problema dado, realizan una serie de comprobaciones sobre dicha solución y generan un veredicto de manera automática. Originalmente se diseñaron para celebrar competiciones de programación, por lo que la retroalimentación que acompaña al veredicto es prácticamente inexistente. A pesar de esta limitación, sistemas como DOMjudge (<https://www.domjudge.org>) o Mooshak (Leal 2003) se llevan usando con mucha aceptación en diversas asignaturas de programación de la Facultad de Informática debido a su robustez y estabilidad. También se han desarrollado jueces automáticos de programación orientados al aprendizaje que mejoran la retroalimentación recibida por los usuarios, lo que les permite comprender sus errores y corregirlos de manera muy autónoma. Entre estas herramientas podríamos destacar *FLOP* (Llana et al. 2014) o *Jutge.org* (Petit et al. 2018), herramientas que también se han utilizado en asignaturas de programación en la UCM y otras universidades.

Las bases de datos son una materia básica en cualquier programa de estudios con perfil tecnológico, y como tal se enseñan en varias de las asignaturas de las facultades de Informática, Ciencias Matemáticas y Estudios Estadísticos. Aunque existen varios modelos de datos para almacenar información, el más común desde los años 70 es el modelo relacional. En este modelo los datos se almacenan en tablas y se consultan usando un lenguaje muy extendido llamado SQL. Este lenguaje es aceptado por la inmensa mayoría de los Sistemas de Gestión de Bases de Datos (SGBD) que existen en la actualidad, de ahí que una de las competencias principales que se persiguen en este tipo de asignaturas sea conocer la sintaxis de SQL y ser capaz de expresar búsquedas mediante consultas SQL. Aunque, estrictamente hablando, SQL es un lenguaje de consulta y manipulación de datos, tiene un gran parecido con otros lenguajes de programación y su aprendizaje es fundamentalmente práctico, por lo que un estudiante necesita resolver una gran cantidad de consultas antes de llegar a dominarlo. Por este motivo, el aprendizaje de SQL se podría beneficiar del uso de jueces automáticos.

Actualmente existen varios sistemas dedicados al aprendizaje de SQL. Por un lado, existen páginas web con tutoriales interactivos de SQL como *SQLBolt* (<https://sqlbolt.com>) o *W3schools* (<https://www.w3schools.com/sql/>), en los que los estudiantes van aprendiendo los aspectos básicos de SQL mediante explicaciones

---

<sup>1</sup> Este trabajo ha sido financiado por el proyecto no. 18 de la convocatoria 2020-21 del programa INNOVA-Docencia de la Universidad Complutense de Madrid

concisas y deben resolver ciertos problemas para poder continuar. También existen jueces automáticos centrados en el aprendizaje de SQL como *SQLab* (García Merayo et al. 2016), *SQL-Tutor* (Mitrovic 2003), *SQLator* (Sadiq, Orłowska y Sadiq 2004), *ActiveSQL* (Russell y Cumming 2004), *SQLify* (Dekeyser, de Raadt y Lee 2007), *aSQLg* (Kleiner, Tebbe y Heine 2013) o *AsseSQL* (Prior 2014). Estos sistemas proponen diferentes ejercicios a los alumnos y utilizan distintas técnicas para comparar su consulta SQL con una consulta correcta proporcionada por los profesores. Dentro de este grupo de herramientas también podríamos destacar *DES* (Sáenz-Pérez 2011), un potente sistema para el aprendizaje de lenguajes de consulta y manipulación de bases de datos que además de SQL incorpora Datalog y álgebra relacional. DES es un sistema interactivo que permite al usuario realizar consultas o modificaciones de la base de datos e inmediatamente ver el resultado. En este sentido, DES es más un sistema para el aprendizaje libre del estudiante, ya que no proporciona ejercicios a resolver ni corrección automática de resultados.

A pesar de la amplia disponibilidad de jueces para el aprendizaje de SQL, hemos detectado en ellos algunas limitaciones que, aunque no impiden aplicarlos ampliamente en la docencia de las bases de datos, reducen su campo de aplicación. La principal limitación de estos sistemas es que se trata de servicios web alojados por terceros, por lo que no es posible asegurar su disponibilidad a lo largo del curso ni se puede tomar ninguna medida de mitigación si ocurre algún incidente puntual. Al no tener poder de decisión sobre estos servicios, tampoco es posible extender la colección de problemas ni eliminar los que se consideran poco adecuados: los ejercicios existentes son los que decidieron sus autores en su momento. Otra limitación importante son los sistemas de gestión de bases de datos (SGBD) soportados. El lenguaje SQL es un estándar, pero su sintaxis y funcionamiento es ligeramente diferente dependiendo del desarrollador del SGBD que se use: Oracle, MySQL, SQLite, PostgreSQL, etc. Aunque manejar estas diferencias es el día a día de un profesional en bases de datos, pueden llegar a ser frustrantes en la fase de aprendizaje. Por ello, es altamente recomendable realizar los ejercicios exactamente en el mismo sistema y versión que se ha utilizado en las clases de teoría. Sin embargo, los sistemas anteriormente mencionados soportan una lista de SGBD prefijada que no es posible ampliar. Si el SGBD que se utiliza en clase tiene soporte en el juez, se podrá usar en la docencia, pero si no está soportado lo más probable es que se prefiera no crear confusión adicional a los estudiantes y se evite usar el sistema. Esto es exactamente lo que ocurre en nuestro caso: el SGBD utilizado en la asignatura “Bases de Datos” de los grados de la Facultad de Informática es Oracle debido a su relevancia en la industria, pero este no está soportado por ningún sistema de los mencionados anteriormente. Por último, otra de las limitaciones es el tipo de problemas soportados. Aunque lo más común es utilizar el lenguaje SQL para realizar consultas, en una base de datos también se pueden definir funciones, procedimientos y disparadores. Estos son conceptos muy relevantes para el uso adecuado de las bases de datos que forman parte de los contenidos de las asignaturas, y sería necesario contar con ejercicios prácticos con corrección automática también para ellos. Sin embargo, los sistemas anteriormente mencionados únicamente soportan problemas sobre consultas SQL y en algunos pocos casos también sobre

sentencias de manipulación de datos, dejando fuera problemas sobre funciones, procedimientos y disparadores.

Debido a las limitaciones anteriores hemos desarrollado LearnSQL, un juez automático diseñado para el aprendizaje de las bases de datos con un ámbito lo más amplio posible: permite ejercicios sobre consultas SQL, pero también proporciona ejercicios donde el estudiante debe definir funciones, procedimientos y disparadores. Durante su diseño y desarrollo se han perseguido los siguientes objetivos:

1. Facilidad de uso tanto para profesores como estudiantes.
2. Aplicación directa en la actual asignatura de «Bases de datos» de la Facultad de Informática, sin esfuerzo adicional por parte del profesorado.
3. Enfoque en el aprendizaje del estudiante, proporcionando toda la información y las ayudas para que entienda el error en su envío y pueda corregirlo.
4. Sistema de código abierto para que cualquier profesor pueda adaptarlo a sus necesidades y desplegarlo en sus equipos.
5. Fomento de la participación e implicación del estudiante mediante ludificación.

En el resto de esta comunicación se presentará el sistema LearnSQL, incluyendo ejemplos de uso. También se comentarán los prometedores resultados obtenidos tras haber usado una versión inicial del sistema en un grupo de la asignatura “Bases de Datos” en la Fac. de Informática durante los meses de septiembre 2020–febrero 2021.

## **El sistema LearnSQL**

Desde el punto de vista tecnológico, LearnSQL se ha desarrollado como una aplicación web en el lenguaje de programación Python utilizando el entorno Django. Todo el código fuente está disponible en <https://github.com/emartinm/lsql/> y es abierto bajo licencia MIT, una de las licencias de código abierto más permisivas que existen. De esta manera cualquier profesor podrá descargar y ejecutar el juez en sus equipos, además de realizar cualquier mejora o modificación del sistema para adaptarlo a sus necesidades. El juez utiliza el SGBD PostgreSQL para almacenar sus datos internos (usuarios, ejercicios, envíos, etc.) mientras que para la corrección de los envíos de los estudiantes utiliza un servidor Oracle 11g externo. Al mantener separados los datos operativos del juez y el componente de corrección de envíos se consigue una mejor protección frente a fallos o envíos de usuarios maliciosos, a la vez que facilita la extensión del sistema de corrección a otros SGBD que se puedan necesitar en el futuro.

Para facilitar el uso del juez por los profesores de la Facultad de Informática en sus grupos, este se ha desplegado en un servidor accesible desde <https://learn.fdi.ucm.es/sql/>. Actualmente no admite registro generalizado, sino que está disponible únicamente para estudiantes de grupos de la asignatura “Base de Datos” de la Facultad de Informática y sus profesores. Los estudiantes pueden acceder a los problemas y realizar envíos, mientras que los profesores también tienen acceso al interfaz de administración donde pueden

crear, borrar, actualizar y consultar todos los elementos del juez: problemas, colecciones, usuarios, envíos, logros...

LearnSQL soporta distintos tipos de ejercicios tratando de dar soporte a los contenidos más usuales de las asignaturas sobre bases de datos. Concretamente soporta los siguientes tipos de problemas:

- Consulta SQL: el estudiante debe construir una consulta SQL que dé respuesta a una pregunta concreta sobre una base de datos fijada.
- Manipulación de datos: en estos ejercicios el usuario debe introducir sentencias SQL para modificar los datos (añadir, borrar o actualizar) de tal manera que la base de datos final refleje los cambios solicitados en el enunciado.
- Definición de funciones PL/SQL: el usuario debe enviar un bloque de código PL/SQL (el lenguaje imperativo nativo de Oracle) que defina una función con un comportamiento dado.
- Definición de procedimientos PL/SQL: similar a los ejercicios sobre funciones, pero en este caso se solicita la definición de un procedimiento que en lugar de devolver un valor realice ciertas modificaciones sobre la base de datos.
- Definición de disparadores: similar a un ejercicio de procedimientos, pero definiendo un disparador, es decir, un fragmento de código que se ejecuta automáticamente cuando ocurre cierta modificación en la base de datos.
- Discriminar consultas: en este tipo de ejercicios el usuario recibe una consulta SQL que es incorrecta pero que devuelve los datos correctos únicamente porque la base de datos de ejemplo mostrada en el enunciado no es suficientemente variada (p.ej. encontrar todos los jugadores de más de 1.5m cuando en la base de datos únicamente hay jugadores de más de 2m). En este caso el usuario debe analizar y comprender la consulta, detectar por qué está devolviendo los datos correctos y modificar la base de datos incorporando esa variedad que hará fallar a la consulta errónea.

Como hemos comentado, los demás jueces para el aprendizaje de bases de datos únicamente soportan ejercicios de las dos primeras categorías, siendo los ejercicios sobre funciones, procedimientos, disparadores y discriminación de consultas una contribución novedosa de LearnSQL. La corrección de los envíos de los usuarios se realiza mediante su ejecución en Oracle 11g y la posterior comparación de los resultados obtenidos frente a la solución oficial proporcionada por el profesor, pudiendo utilizarse varios casos de prueba diferentes para corregir un envío. La corrección mediante ejecución es la opción más usual en este tipo de jueces, aunque hay otros enfoques como comprobar la equivalencia semántica de la consulta del usuario y la solución oficial (Sadiq, Orłowska y Sadiq 2004, Dekeyser, de Raadt y Lee 2007) o realizar comprobaciones estáticas sobre el texto de la solución enviada.

### Colecciones de problemas

Nombre	Resueltos	Total problemas
<a href="#">Consultas SELECT básicas</a>	4	12
<a href="#">Reuniones de tablas y operaciones de conjuntos</a>	0	8
<a href="#">Agregaciones y agrupaciones</a>	0	12
<a href="#">Consultas anidadas</a>	1	5
✓ <a href="#">Funciones PL/SQL</a>	4	4
✓ <a href="#">Procedimientos PL/SQL</a>	2	2

Figura 1. Listado de colecciones disponibles en LearnSQL.

### Consultas SELECT básicas

Consultas SELECT involucrando una única tabla

#### Listado de problemas

Nombre	Número de envíos	1°	2°	3°
✓ <a href="#">Tabla completa</a>	103	Ana	Juan	Eva
<a href="#">Filtrar columnas</a>	3	Raúl	Soledad	-
<a href="#">Seleccionar filas</a>	0	-	-	-

Figura 2. Listado de los problemas de una colección.

LearnSQL muestra los problemas organizados en colecciones tal y como se aprecia en la figura 1. Cada colección contiene un conjunto de problemas de temática similar, y para cada una se muestra el número total de problemas resueltos por el usuario actual y el total de problemas disponibles. Cuando un usuario ha resuelto todos los problemas de una colección, esta se decora con una marca verde. El listado de problemas de una colección es similar, tal y como se puede ver en la figura 2. En este caso, además, se muestra el número de envíos del usuario actual a cada ejercicio y el podio de los tres primeros usuarios que han resuelto cada problema. El podio aporta una dosis de ludificación que esperamos motive a los estudiantes a intentar más ejercicios.

Una vez se selecciona un problema se accede a su enunciado, como se aprecia en la figura 3. En este caso se trata de un ejercicio de consulta SQL. En primer lugar, se muestra el título y el texto del ejercicio, que es configurable por el profesor. A continuación, se incluye el conjunto de tablas y filas sobre las que se probará la consulta del usuario, seguido del resultado esperado en ese escenario.

**Tabla completa** ✓

Considera una tabla que almacena algunos datos sobre clubes de fútbol definida de la siguiente manera:

```
CREATE TABLE Club(
  CIF CHAR(9) PRIMARY KEY,
  Nombre VARCHAR2(40) NOT NULL UNIQUE,
  Sede VARCHAR2(30) NOT NULL,
  Num_Socios NUMBER(10,0) NOT NULL,
);
```

Escribe una consulta SQL que devuelve **todo el contenido** de la tabla `club`. El esquema del resultado debe ser el siguiente:

```
(CIF, Nombre, Sede, Num_Socios)
```

Base de datos [Descargar script](#)

CLUB [+](#)

CIF	NOMBRE	SEDE	NUM_SOCIOS
-----	--------	------	------------

**Resultado esperado**

CIF	NOMBRE	SEDE	NUM_SOCIOS
11111111X	Real Madrid CF	Concha Espina	70000
11111112X	Futbol Club Barcelona	Aristides Maillol	80000
11111113X	Paris Saint-Germain Football Club	Rue du Commandant Guilbaud	1000

Figura 3. Enunciado de un problema de consulta SQL.

Bajo el resultado esperado aparece un cuadro de texto con resultado de sintaxis SQL en la que el usuario debe introducir la solución (también existe la posibilidad de cargar la solución a partir de un fichero de texto previamente creado). Cada vez que un usuario realiza un envío, LearnSQL lo revisa y produce un veredicto junto con una retroalimentación detallada en caso de encontrar un error. Entre los diferentes tipos de veredicto podemos destacar:

- **Aceptado:** la solución enviada ha cumplido todas las comprobaciones.
- **Error en ejecución:** la solución ha fallado al ser ejecutada por Oracle, p. ej. la sintaxis es incorrecta o se hace referencia a tablas que no existen.
- **Resultados incorrectos:** el envío se ha ejecutado sin problema, pero los resultados obtenidos no son los esperados.

Como uno de los objetivos perseguidos es que el juez facilite al máximo el aprendizaje, cuando un envío recibe un veredicto de error, este viene acompañado de una retroalimentación detallada. En el caso de errores de compilación, se muestra la línea y columna exacta y el mensaje de error generado por Oracle. En el caso de resultados

incorrectos, se incluye una comparación pormenorizada de las diferencias detectadas entre los resultados obtenidos por la solución del usuario y la solución oficial. En la figura 4 se puede ver un ejemplo de este tipo de retroalimentación, donde se destaca la fila que no devuelve la solución del estudiante.

**Retroalimentación**

Faltan algunas filas que deberían aparecer.

Resultado generado por tu código:

CIF	NOMBRE	SEDE	NUM_SOCIOS
11111111X	Real Madrid CF	Concha Espina	70000
11111112X	Futbol Club Barcelona	Aristides Maillol	80000

Filas que faltan:

CIF	NOMBRE	SEDE	NUM_SOCIOS
11111113X	Paris Saint-Germain Football Club	Rue du Commandant Guilbaud	1000

Figura 4. Retroalimentación obtenida en un ejercicio de consulta SQL con solución errónea.

Un aspecto novedoso de LearnSQL frente a otros sistemas para el aprendizaje de bases de datos es que incorpora elementos de ludificación con la intención de motivar a los estudiantes. Además de mostrar los podios por cada problema, LearnSQL dispone de un sistema de logros que los estudiantes pueden obtener según van resolviendo problemas. El número de logros y su definición son configurables por los profesores, p.ej. “resolver 10 problemas de consultas SQL” o “resolver 5 problemas de la colección sobre funciones”. Cada usuario tiene una página de logros pública que todos pueden ver donde se lista qué logros ha obtenido y en qué momento, además de consultar qué otros logros no se han conseguido. Por otro lado, LearnSQL también genera clasificaciones de los distintos estudiantes por cada colección. Para ello ordena los estudiantes del mismo grupo de clase en base al número de problemas resueltos en esa colección, y en caso de empate penaliza a aquellos estudiantes que han necesitado más envíos para resolver dichos problemas. Tal y como se ve en la figura 5, por cada usuario y problema el ranking muestra una casilla con formato “C/T (P)” con el número de envíos correctos (C), el número de envíos totales (T) y el primer envío correcto recibido (P). Nótese que, para maximizar la visibilidad de los logros obtenidos por cada estudiante, esta información se muestra al lado del nombre de usuario junto con un enlace que lleva a su página de logros.

Pos.	Usuario	EJ1	EJ2	EJ3	EJ4	EJ5	Puntuación	Resueltos
1	Ana 🏆 x4	2/10 (7)	1/4 (4)	1/4 (4)	2/3 (2)	1/1 (1)	18	5
2	Juan 🏆 x5	1/17 (17)	1/3 (3)	1/3 (3)	1/7 (7)	1/1 (1)	31	5
3	Isabel 🏆 x5	1/26 (26)	1/10 (10)	1/2 (2)	1/3 (3)	2/2 (1)	42	5
4	José 🏆 x4	1/1 (1)	1/12 (12)	1/4 (4)	1/23 (23)	1/5 (5)	45	5
5	Ivette 🏆 x2	1/140 (140)	3/50 (48)	0/32 (32)	1/13 (13)	0/12 (12)	201	3

Figura 5. Clasificación de una colección de 5 ejercicios.

Por último, aunque el idioma principal de LearnSQL es el castellano, el sistema tiene soporte multidioma. Actualmente la interfaz está disponible en español e inglés y el usuario puede cambiar a conveniencia mediante un selector. También permite que haya problemas en distintos idiomas indicando gráficamente mediante una bandera el idioma de los problemas que no están en español. De esta manera, LearnSQL se puede utilizar en grupos con docencia en inglés e incluso en grupos con un alto número de estudiantes Erasmus.

## Evaluación preliminar

Durante el primer cuatrimestre del curso 2020-21 se ha utilizado una versión inicial de LearnSQL sin elementos de ludificación en un grupo de la asignatura “Bases de Datos” de la Facultad de Informática. Para ello, se crearon usuarios para todos los estudiantes del grupo y se publicó una serie de problemas donde la inmensa mayoría trataban sobre consultas SQL. El uso del juez era completamente opcional por parte de los alumnos y se les presentó como una herramienta con la que podrían practicar en cualquier momento, incluyendo los laboratorios prácticos de la asignatura. La aceptación de los estudiantes ha sido bastante alta: de los 51 estudiantes matriculados, 26 de ellos han realizado envíos. En total, a lo largo del periodo de septiembre de 2020 a febrero de 2021 se recibieron 5.517 envíos, resultando en una media de 212,2 envíos por usuario. Al finalizar el curso se pidió a los estudiantes que rellenaran una encuesta de valoración anónima a través del Campus Virtual, que recibió 11 envíos de estudiantes que reconocieron haber resuelto más de 20 problemas en LearnSQL. Esta encuesta constaba de varias preguntas de puntuación numérica en el rango 1-5 y algunos campos de comentarios abiertos para indicar los puntos fuertes y débiles del juez. Entre las preguntas numéricas podríamos destacar las siguientes, junto con su media:

- La organización de las distintas páginas del juez es adecuada: 4.73.
- El juez es fácil de usar: 4.82.
- La retroalimentación que muestra el juez cuando un envío no es correcto es informativa: 2.73.
- El juez automático te ha ayudado a aprender a realizar consultas SQL, en proporción al uso que has hecho de él: 4.82.

Como se aprecia, los estudiantes reconocen que el juez les ayudó mucho a aprender a realizar consultas SQL, a la vez que destacan su facilidad de uso. Por otro lado, la retroalimentación obtenida no resulta completamente satisfactoria. Desde febrero se han incorporado mejoras y correcciones a este apartado, como indicar la línea y columna en los errores de ejecución, así que estamos esperando al próximo curso para validar si esta información es suficiente. Con respecto a las preguntas de respuesta abierta, las contestaciones incidían principalmente en la mejora de la retroalimentación, además de ampliar el conjunto de problemas. En general, el resultado de esta experiencia ha sido muy prometedor y planeamos seguir utilizando LearnSQL el próximo curso con la esperanza de obtener una aceptación aún más alta.

## Conclusiones y trabajo futuro

En esta comunicación hemos presentado LearnSQL, un juez automático para el aprendizaje de bases de datos que hemos desarrollado dentro del proyecto no. 18 de la convocatoria INNOVA-Docencia del curso 2020-21. A diferencia de otros sistemas similares, LearnSQL es completamente libre y cualquiera puede acceder a su código fuente en <https://github.com/emartinm/lsql/> para usarlo o adaptarlo a sus necesidades. Este sistema también presenta como novedades el soporte de una variedad de problemas diferentes sobre SQL, concretamente los problemas sobre PL/SQL (funciones, procedimientos y disparadores), además de los problemas de discriminar consultas. Aparte de proporcionar información detallada de los errores detectados en los envíos de los estudiantes para facilitar el aprendizaje, LearnSQL también incorpora elementos de ludificación que esperamos motiven e involucren a los estudiantes en el proceso de aprendizaje. Desde un punto de vista práctico, hemos aplicado una versión inicial del juez en un grupo de la asignatura “Bases de Datos” de la Facultad de Informática y los resultados son esperanzadores, tanto a nivel de participación como valoración de los estudiantes.

Entre las mejoras y extensiones que consideramos para el juez, una de las más relevantes es ampliar el conjunto de problemas disponibles, que actualmente está en torno a 45, con una gran presencia de ejercicios sobre consultas SQL. También valoramos extender el componente de corrección para otros SGBD además de Oracle, como PostgreSQL. Cabe destacar que este componente no está ligado a bases de datos relacionales, sino que se podrían abarcar otros modelos de datos NoSQL como documentos (MongoDB), clave-valor (Redis) o incluso grafos (Neo4J).

*Agradecimientos.* A Iker Burgoa, Tamara Huertas, Daniel Ibáñez e Iván Ruiz por sus contribuciones a LearnSQL como parte de su Trabajo Fin de Grado.

## Referencias bibliográficas

Dekeyser, Stijn y de Raadt, Michael y Lee, Tien Yu. 2007. "Computer assisted assessment of SQL query skills". En *Proc. 18th Australasian Database Conference (ADC 2007)* 63, 53-62. Australian Computer Society.

García Merayo, María de las Mercedes y Núñez García, Manuel y Pareja Flores, Cristobal y García Ruiz, Yolanda y Estévez Martín, Sonia y Martínez Torres, Rafael y Camacho Magriñán, María Azahara y Cerro Cañizares, Pablo y Marco González, Juan Carlos. "SQLab: Laboratorio virtual lenguaje SQL". Última modificación el 19 de enero de 2021. <https://eprints.ucm.es/id/eprint/45925/>

Kleiner, Carsten y Tebbe, Christopher y Heine, Felix. 2013. "Automated grading and tutoring of SQL statements to improve student learning". En *Proc. 13th Koli Calling International Conference on Computing Education Research*, 161-168. ACM.

Llana, Luis y Martin-Martin, Enrique y Pareja-Flores, Cristóbal y Velázquez-Iturbide, J. Ángel. 2014. "FLOP: A User-Friendly System for Automated Program Assessment". *Journal of Universal Computer Science* 20, no. 9: 1304-1326.

Leal, José Paulo; Silva, Fernando. 2003. "Mooshak: A Web-based multi-site programming contest system". *Software: Practice and Experience* 33, no. 6: 567-581.

Mitrovic, Antonija. 2003. "An intelligent SQL tutor on the web". *International Journal of Artificial Intelligence in Education* 13, no. 2-4: 173-197.

Petit, Jordi y Roura, Salvador y Carmona, Josep y Cortadella, Jordi y Duch, Amalia y Giménez, Omer y Mani, Anaga y Mas, Jan y Rodríguez-Carbonell, Enric y Rubio, Albert y de San Pedro, Javier y Divya Venkataramani. 2018. "Jutge.org: Characteristics and Experiences". *IEEE Trans. Learn. Technol* 11, no. 3: 321-333. IEEE.

Prior, Julia R. 2014. "AsseSQL: an online, browser-based SQL skills assessment tool". En *Proc. 2014 conference on Innovation & technology in computer science education (ITiCSE 2014)*, 327-327. ACM.

Russell, Gordon y Cumming, Andrew. 2004. "Improving the Student Learning Experience for SQL Using Automatic Marking". En *Proc. 2004 IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2004)*, 281-288.

Sadiq, Shazia y Orłowska, Maria y Sadiq, Wasim y Lin, Joe. 2004. "SQLator: an online SQL learning workbench". En *Proc. 9th SIGCSE conference on Innovation and technology in computer science education (SIGCSE 2004)*, 223-227. ACM.

Sáenz-Pérez, Fernando. 2011. "DES: A deductive database system". *Electronic notes in theoretical computer science* 271: 63-78.