

TutNorBD: Assistant for teaching and learning process of relational database normalization up to 3NF from a universal table

Danilo Mendoza

*Departamento de Ciencias de la Computación y Electrónica
Universidad Técnica Particular de Loja
Loja, Ecuador
odmendoza@utpl.edu.ec*

Nelson Piedra

*Knowledge Based Systems Research Group
Universidad Técnica Particular de Loja
Loja, Ecuador
nopiedra@utpl.edu.ec*

Abstract—Relational database normalization is a formal data organization process that seeks to make the database more flexible by eliminating data redundancy and inconsistencies. Informally, a relational database relationship is often described as "normalized" if it conforms to the third normal form. Most relationships in 3NF are free of insert, update, and delete anomalies. In this work the design of a normalization assistant is presented. The purpose of the presented proposal is to support the teaching and learning process of relational database normalization, making the user take his universal data table to third normal form in an agile and simple way.

Index Terms—normalization, relational database, normal forms, assistant

I. INTRODUCCIÓN

La normalización de bases de datos relacionales [11] es un proceso que consiste en designar y aplicar una serie de reglas a las relaciones, con objeto de minimizar la redundancia de datos, facilitando su posterior administración [10]. El proceso de normalización incluye crear tablas y establecer relaciones entre esas tablas de acuerdo con reglas diseñadas tanto para proteger los datos, como para hacer que la base de datos sea más flexible al eliminar la redundancia y la dependencia inconsistente [1] [6].

Los datos redundantes desperdician espacio en disco y capacidad de procesamiento, pero sobre todo crean problemas de mantenimiento. Si los datos existen en más de un lugar, para mantener la integridad de la base de datos, en cada transacción estos deben insertarse, actualizarse o eliminarse en todas las ubicaciones donde se encuentran replicados. Esto no es aconsejable sobre todo en sistemas transaccionales ya que incrementa la latencia. Debido a esto, por ejemplo, para realizar un cambio de teléfono del cliente en un sistema de ventas es mucho más fácil de implementar si estos datos se almacenan solo en la tabla Clientes y en ningún otro lugar de la base de datos. Por otro lado, hay que tener en cuenta que las dependencias inconsistentes pueden dificultar el acceso a los datos porque la ruta para encontrar los datos puede faltar o estar rota.

Hay algunas reglas para la normalización de la base de datos. Cada regla se llama "forma normal". Si se observa la

primera regla, se dice que la base de datos está en "primera forma normal". Si se observan las tres primeras reglas, se considera que la base de datos está en "tercera forma normal". Aunque son posibles otros niveles de normalización, la tercera forma normal se considera el nivel más alto necesario para la mayoría de las aplicaciones.

Al igual que con muchas reglas y especificaciones formales, los escenarios del mundo real no siempre permiten un cumplimiento perfecto. En general, la normalización requiere tablas adicionales y algunos clientes encuentran esto engorroso. La no observación de una de las tres primeras reglas de normalización provocará datos redundantes y dependencias inconsistentes.

En consecuencia, en el diseño de una base de datos relacional la normalización es un aspecto fundamental; sin embargo, los estudiantes suelen tener problemas para entender los conceptos y aplicarlos a casos reales, ya que suele ser una tarea compleja sobre todo si se trata de una gran cantidad de datos.

Frente a esta necesidad, en este trabajo se presenta el diseño de un asistente de normalización. El objetivo es apoyar el proceso de enseñanza - aprendizaje de normalización de bases de datos relacionales, haciendo que el usuario lleve a tercera forma normal su tabla universal de datos de una manera ágil y sencilla. La implementación de este asistente se ha realizado en el lenguaje de programación Python [13], se utiliza la librería para análisis de datos Pandas [14] y el microframework Flask [15] para desarrollo web. El documento está estructurado de la siguiente manera, se analizan trabajos relacionados, se describe el proceso de normalización de bases de datos relacionales, a continuación se presenta el diseño de la propuesta, la implementación y pruebas y finalmente se presentan las conclusiones.

II. TRABAJOS RELACIONADOS

A continuación se describen algunos trabajos relacionados en los que se describen herramientas que se han desarrollado para el proceso de enseñanza - aprendizaje de normalización de Bases de datos, así:

- JMathNorm, es una herramienta para la normalización de bases de datos relacionales (RDB) usando Mathematica, que incluye 1FN, 2FN, 3FN y Boyce-Codd (BCNF). Es una herramienta completa que se puede utilizar para el diseño de bases de datos en tiempo real, además de ayudar a enseñar conceptos fundamentales de normalización de DB. Está escrito en Java y utiliza JLink para manejar el kernel de Mathematica [1].

- En [2] se presenta un nuevo método de normalización de base de datos relacional automatizado y completo. Produciendo la matriz de dependencia y la matriz gráfica dirigida, y de ahí genera las formas normales 2NF, 3NF y BCNF. Todas las tablas se generan a medida que avanza el procedimiento. Además, distingue automáticamente una clave primaria para cada tabla final que se genera.

- RDBNorma, que utiliza un enfoque novedoso para representar un esquema de base de datos relacional y sus dependencias funcionales en la memoria de la computadora utilizando solo una lista vinculada y utilizada para semiautomatizar el proceso de normalización del esquema de base de datos relacional hasta 3FN. Presenta los algoritmos para convertir una relación en la segunda y tercera forma normal mediante el uso de la representación anterior [3].

- Intelligent Tutoring System for Database Normalization, desarrollado por un grupo de docentes, es un tutor inteligente, que enseña a los estudiantes el proceso de normalización y los ayuda a normalizar las bases de datos [4].

- En [5], se propone una nueva herramienta de aprendizaje permite a los estudiantes ver paso a paso cómo llevar a cabo cada forma normal hasta el 3NF. Esta herramienta se basa en el análisis de datos de la tabla de entrada, y no en el análisis del esquema o las dependencias iniciales definidas por las reglas de negocio. Desarrolla nuevos algoritmos para procesar la tabla de entrada (definida por el usuario), así como las tablas creadas durante el proceso de normalización.

III. LA NORMALIZACIÓN

En esta sección se explica la terminología básica utilizada en el proceso de normalización. Una comprensión básica de esta terminología es útil cuando se discute el diseño de una base de datos relacional.

Codd introdujo el concepto de normalización y lo que ahora se conoce como la primera forma normal (1NF) en 1970. Codd pasó a definir la segunda forma normal (2NF) y la tercera forma normal (3NF) en 1971, y Codd y Raymond F. Boyce definieron la forma normal Boyce-Codd (BCNF) en 1974.

A. Dependencias funcionales

Comprender adecuadamente el concepto de dependencia funcional constituye la base para el proceso de normalización. Las dependencias funcionales describen las relaciones entre los atributos de una relación. Una dependencia funcional es una restricción entre dos conjuntos de atributos de un base de datos. Una dependencia funcional, denotada por $X \rightarrow Y$, entre dos conjuntos de atributos X y Y que pertenecen a la relación R especifica una restricción sobre las posibles tuplas

que pueden formar un estado de relación r en R . La restricción es que, para cualquiera de las dos tuplas t_1 y t_2 en r que tienen $t_1[X] = t_2[X]$, también deben tener $t_1[Y] = t_2[Y]$. Esto quiere decir que los valores de Y dependen, o son determinados por, los valores de X . Y al revés, que los valores de X determinan funcionalmente a los valores de Y [12].

En otras palabras, X determina funcionalmente a Y en un esquema de relación R si, y solo si, siempre que dos tuplas de r (R) concuerden, el valor de X necesariamente debe concordar con el valor de Y . En la Figura 1 se observa la dependencia funcional $X \rightarrow Y$, ya que siempre que X es igual a 'a', Y es igual a 'b'.

X	Y	Z
a	b	c
c	d	e
a	b	e

Fig. 1. Dependencia funcional $X \rightarrow Y$

B. Claves, superclaves, claves candidatas, clave primaria y claves secundarias

Si una restricción en R establece que no puede haber más de una tupla con un valor X dado en cualquier instancia de relación r (R), entonces, X es una *clave* de R , esto implica que $X \rightarrow Y$ para cualquier subconjunto de atributos Y de R (porque la restricción *clave* implica que no hay dos tuplas en ningún estado válido r (R) que tendrán el mismo valor de X). Si X es una *clave* de R , entonces $X \rightarrow R$ [12].

Una *superclave* de un esquema de relación $R = \{A_1, A_2, \dots, A_n\}$ es un conjunto de atributos $S \subseteq R$ con la propiedad de que no hay dos tuplas t_1 y t_2 en ningún estado de relación válido r de R tendrán $t_1[S] = t_2[S]$. Una *clave* K es una *superclave* con la propiedad adicional de que la eliminación de cualquier atributo de K hará que K no sea una *superclave* [12].

La diferencia entre una clave y una superclave es que una clave debe ser mínima; es decir, si tenemos una clave $K = \{A_1, A_2, \dots, A_k\}$ de R , entonces $K - \{A_i\}$ no es una clave de R para ningún A_i , $\{1 \leq i \leq k\}$ [12].

Si un esquema de relación tiene más de una clave, cada una se denomina *clave candidata*. Una de las claves candidatas se designa como *clave primaria*, y las otras se denominan *claves secundarias*. En una base de datos relacional práctica, cada esquema de relación debe tener una clave primaria. Si no se conoce una clave candidata para una relación, toda la relación se puede tratar como una superclave predeterminada [12].

C. Tipos de dependencias funcionales

Ahora bien, existen tres tipos de dependencias funcionales, a saber, dependencia funcional completa, dependencia funcional parcial y dependencia funcional transitiva.

Existe *dependencia funcional completa* cuando todos los valores de lado izquierdo de la flecha determinan funcionalmente a todos los valores del lado derecho de la flecha. Cuando

todos los atributos no claves de la relación dependen de forma funcional y completa de la clave primaria [12]. Por ejemplo:

$$\begin{aligned} PK &= \{X, Y\} \\ \{X, Y\} &\rightarrow Z \\ \{X, Y\} &\rightarrow \{Z, W\} \end{aligned}$$

Existe *dependencia funcional parcial* cuando solo parte del lado izquierdo de la flecha es necesaria para determinar a todos o a alguno de los valores del lado derecho de la flecha. Cuando existen atributos no clave que dependen de parte de la superclave [12]. Este tipo de dependencia funcional solo existe si la clave primaria es una superclave compuesta (formada por 2 o más atributos). Por ejemplo:

$$\begin{aligned} PK &= \{X, Y\} \\ \{X, Y\} &\rightarrow Z \\ X &\rightarrow Z \end{aligned}$$

Existe *dependencia funcional transitiva* cuando un atributo no clave determina funcionalmente a otro atributo no clave, o viceversa, cuando un atributo no clave es determinado por otro atributo no clave y no por la clave primaria o superclave [12]. Por ejemplo:

$$\begin{aligned} PK &= X \\ X &\rightarrow Y \\ Y &\rightarrow Z \\ X &\rightarrow Z \text{ (Dependencia funcional transitiva)} \end{aligned}$$

D. Normalización

La normalización de datos consiste en un proceso de análisis de los esquemas de relación dados en función de sus dependencias funcionales y claves primarias con el fin de minimizar la redundancia y las anomalías a la hora de la inserción, eliminación y actualización de la base de datos [7]. De esta forma, la normalización es un proceso mediante el cual se toma un diseño de base de datos existente con algunas anomalías y redundancias, y se mejora descomponiendo relaciones, en base a un conjunto conocido de dependencias funcionales [1].

Ahora presentamos las tres primeras formas normales: 1NF, 2NF y 3NF. Estos fueron propuestos por Codd en 1972 como una secuencia para lograr el estado deseable de relación 3NF progresando a través de los estados intermedios de 1NF y 2NF si fuera necesario. Como veremos, 2NF y 3NF atacan independientemente diferentes tipos de problemas derivados de dependencias funcionales problemáticas entre atributos. Sin embargo, por razones históricas, es costumbre seguirlos en esa secuencia; por lo tanto, por definición, una relación 3NF ya satisface a 2NF, y una relación 2NF satisface a 1NF [7].

Existen además la forma normal Boyce-Codd (BCNF), cuarta forma normal (4NF), quinta forma normal (5FN) y forma normal dominio-clave (DKNF). Sin embargo, en la mayoría de los casos, transformar nuestra base de datos a 3NF suele ser suficiente para evitar anomalías de inserción, eliminación y actualización, ya que las tres primeras formas normales contemplan los problemas más comunes relacionados con la redundancia de datos [7].

IV. DISEÑO DE LA PROPUESTA

A. Funcionamiento General

La herramienta que proponemos permite cargar un archivo Excel, en formato *.xls* o *.xlsx*, que contiene la tabla universal que el usuario desea normalizar. Nuestra propuesta (a) permite llevar la tabla universal a 1FN solicitando al usuario que seleccione la clave primaria y los atributos no clave que contienen datos no atómicos; (b) permite llevar las tablas desde 1FN a 2FN solicitando al usuario que, en caso de existir, seleccione los atributos en los que existe dependencia funcional parcial; y, (c) permite llevar tablas desde 2FN a 3FN solicitando al usuario que, en caso de existir, seleccione los atributos no clave que tengan dependencia funcional transitiva.

B. Diagrama de los Casos de Uso implementados

En base a los requerimientos mencionados en el apartado anterior, se ha identificado 4 casos de uso, como se observa en la Figura 2.

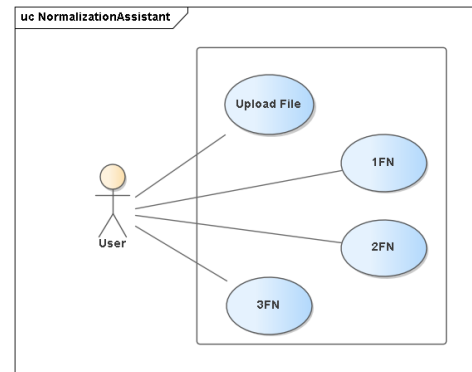


Fig. 2. Diagrama de Casos de Uso.

1) *Caso de Uso: Cargar Archivo*: Es lo primero que el usuario realiza y esto le permite aplicar la normalización a cualquier tabla universal que desee. Ver especificación de este caso de uso en la Tabla I.

TABLE I
ESPECIFICACIÓN DEL CASO DE USO CARGAR ARCHIVO

Nombre del Caso de uso	Cargar Archivo
Actores participantes	Iniciador por: Usuario
Flujo de eventos	<ul style="list-style-type: none"> - Este caso de uso empieza cuando el Usuario selecciona la opción 'Cargar archivo'. - El sistema despliega un formulario para seleccionar el archivo desde equipo del usuario. - El usuario selecciona el archivo Excel que contiene la tabla universal a ser normalizada y da clic en 'Subir'. - El sistema carga el archivo desde el equipo del usuario. Termina el caso de uso.
Condiciones de entrada	Ninguna
Condiciones de salida	El usuario debe haber seleccionado un archivo Excel válido en los formatos <i>.xls</i> o <i>.xlsx</i> que contenga la tabla universal.

2) *Caso de Uso: Primera Forma Normal:* La primera forma normal (1NF) ahora se considera parte de la definición formal de una relación en el modelo relacional básico (plano); Históricamente, se definió para no permitir atributos de valores múltiples, atributos compuestos y sus combinaciones. Establece que el dominio de un atributo debe incluir solo valores atómicos (simples, indivisibles) y que el valor de cualquier atributo en una tupla debe ser un valor único del dominio de ese atributo. Por lo tanto, 1NF no permite tener un conjunto de valores, una tupla de valores o una combinación de ambos como valor de atributo para una sola tupla. En otras palabras, 1NF no permite relaciones dentro de relaciones o relaciones como valores de atributo dentro de tuplas. Los únicos valores de atributo permitidos por 1NF son valores atómicos únicos (o indivisibles) [7].

Una tabla está en primera forma normal si tiene las siguientes propiedades:

- No tiene celdas con más de un valor
- Todos los valores de cada columna tienen el mismo dominio o tipo de dato (enteros, fechas, cadenas, ...)
- No existen grupos duplicados de datos o registros
- Cada columna tiene un título único que denota el nombre del campo. Si una columna no tiene título, se considera parte de un campo multivalor cuyo nombre es el de la primera columna con nombre que se encuentre a la izquierda.
- La tabla debe tener una clave primaria que puede estar compuesta por una o más columnas. El valor de la clave primaria de cada registro debe ser único.

A continuación, la Tabla II muestra la especificación de este caso de uso.

TABLE II
ESPECIFICACIÓN DEL CASO DE USO PRIMERA FORMA NORMAL

Nombre del Caso de uso	1FN
Actores participantes	Iniciador por: Usuario
Flujo de eventos	<ul style="list-style-type: none"> - Este caso de uso empieza cuando el Usuario selecciona la opción 'Subir' del formulario de carga de archivo del caso de uso Upload File. - El sistema despliega la tabla universal y un formulario que pide al usuario que seleccione los atributos que forman parte de la clave primaria. - El usuario selecciona los atributos que forman parte de la clave primaria y presiona el botón 'Empezar'. - El sistema despliega un nuevo formulario que pide al usuario que seleccione los atributos no clave que contienen datos no atómicos y el separador de los datos no atómicos. - El usuario selecciona los atributos no clave que contienen datos no atómicos y el separador de los datos no atómicos presiona el botón 'Primera Forma Normal'. - El sistema procesa los datos y devuelve la o las tablas resultantes en 1FN. Termina el caso de uso.
Condiciones de entrada	Un archivo Excel que contenga la tabla universal debe estar cargado.
Condiciones de salida	La o las tablas resultantes deben estar en 1FN.

3) *Caso de Uso: Segunda Forma Normal:* Una relación R está en 2NF si cada atributo no clave A en R depende completamente de la clave primaria de R. Una tabla está en segunda forma normal si tiene las siguientes propiedades:

- Está en Primera Forma Normal.
- Todos los atributos no clave dependen de forma funcional y completa de la clave primaria o superclave. O también, ningún atributo no clave tiene dependencia parcial respecto de la clave primaria.
- Los atributos que dependen funcionalmente de una parte de la superclave, serán removidos a otra tabla compuesta por el subconjunto de atributos de la clave primaria y los atributos que dependen de este subconjunto. La nueva tabla tendrá como clave foránea el subconjunto de atributos de la clave primaria y una cardinalidad de uno a muchos.

A continuación, la Tabla III muestra la especificación de este caso de uso.

TABLE III
ESPECIFICACIÓN DEL CASO DE USO SEGUNDA FORMA NORMAL

Nombre de Caso de uso	2FN
Actores participantes	Iniciador por: Usuario
Flujo de eventos	<ul style="list-style-type: none"> - Este caso de uso empieza cuando el Usuario selecciona la opción 'Primera Forma Normal' del formulario de datos atómicos del caso de uso 1FN. - El sistema muestra la o las tablas en 1FN y despliega un formulario para que el usuario seleccione, en caso de existir, la clave primaria parcial de la que dependen otros atributos y los atributos que dependen funcionalmente de la clave primaria parcial antes seleccionada - El usuario selecciona la clave primaria parcial de la que dependen otros atributos no clave y los atributos que dependen funcionalmente de la clave primaria parcial antes seleccionada y presiona el botón 'Continuar' - El sistema procesa los datos y devuelve la o las tablas resultantes en 2FN. El caso de uso termina
Condiciones de entrada	La o las tablas entrantes deben estar en 1FN
Condiciones de salida	La o las tablas resultantes deben estar en 2FN.

4) *Caso de Uso: Tercera forma normal:* Una relación R está en 3NF si satisface 2NF y ningún atributo no clave de R depende transitivamente de la clave primaria. Una tabla está en tercera forma normal si tiene las siguientes propiedades:

- Está en Segunda Forma Normal
- No existen dependencias transitivas, esto es, no existen atributos no clave que dependan funcionalmente de otros atributos no clave [7].
- Los atributos que dependan funcionalmente de otros atributos no clave serán removidos a otra tabla compuesta por los atributos no clave determinantes y los que dependen de estos. La nueva tabla tendrá como clave foránea al atributo determinante y una cardinalidad de uno a muchos.

La Tabla IV muestra la especificación de este caso de uso.

TABLE IV
ESPECIFICACIÓN DEL CASO DE USO TERCERA FORMA NORMAL

Nombre de Caso de uso	3FN
Actores participantes	Iniciador por: Usuario
Flujo de eventos	<ul style="list-style-type: none"> - Este caso de uso empieza cuando el Usuario selecciona la opción 'Continuar' del formulario de clave primaria parcial del caso de uso 2FN - El sistema muestra la o las tablas en 2FN y despliega un formulario para que el usuario seleccione, en caso de existir, los atributos determinantes y los atributos que tienen dependencia funcional transitiva - El usuario selecciona los atributos determinantes y los atributos que tienen dependencia funcional transitiva y presiona el botón 'Continuar' - El sistema procesa los datos y devuelve la o las tablas resultantes en 3FN. Termina el caso de uso.
Condiciones de entrada	La o las tablas entrantes deben estar en 2FN
Condiciones de salida	La o las tablas resultantes deben estar en 3FN.

C. Diagrama de flujo

Como se ha visto hasta ahora, el proceso de normalización es secuencial. Después de cargar el archivo inicial, las tablas no pueden pasar a Segunda Forma Normal mientras no se elimine la dependencia funcional parcial, ni se puede pasar a Tercera Forma Normal si existe dependencia funcional transitiva. Ver Figura 3.

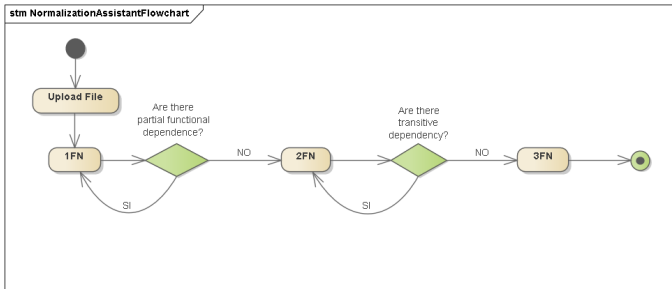


Fig. 3. Diagrama de flujo.

D. Diagrama despliegue

La arquitectura de la aplicación es de tipo cliente-servidor. El usuario se conecta desde el navegador de su equipo al servidor Heroku donde se encuentra desplegada la aplicación. Ver Figura 4.

V. IMPLEMENTACIÓN Y PRUEBAS

A. Desarrollo

La aplicación propuesta es de tipo web. Ha sido desarrollada en el lenguaje de programación Python en su versión 3.7.3 [13], utilizando el framework para crear aplicaciones web llamado Flask [14]. Se ha utilizado, además, la librería Pandas de análisis de datos para el manejo más eficiente de los datos [15]. Finalmente, para el despliegue de la aplicación se ha

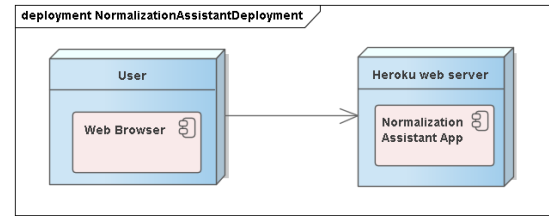


Fig. 4. Diagrama de Despliegue.

utilizado la plataforma de computación en la nube, Heroku [16].

El código fuente de la aplicación se encuentra disponible en un repositorio de GitHub en el enlace siguiente: https://github.com/odmendoza/basic_normalizer.

La aplicación desplegada en Heroku se encuentra disponible en el siguiente enlace: <https://basic-normalizer.herokuapp.com/>.

B. Interfaz de Usuario

El asistente de normalización inicia solicitando al usuario que cargue un archivo Excel en formato .xls o .xlsx donde se aloja la tabla universal de la base de datos que se desea normalizar, como se observa en la Figura 5.



Fig. 5. Interfaz de carga de archivo. El usuario carga la tabla universal desde su equipo en un archivo Excel.

Una vez seleccionado el archivo, el sistema lo carga y muestra al usuario la tabla universal, tal como se observa en la Figura 6.

Como ejemplo, para esta demostración, utilizaremos una tabla que registra los datos de un estudiante y la materia que éste ha tomado en un determinado periodo académico. La estructura de la tabla es como sigue:

Enrollment-history (name , lastName ,birthDate, address, city, phone, subjectCode , subjectTitle, room, capacity, semester , year)

A continuación, se pide al usuario que ingrese el o los atributos de la clave primaria, ver Figura 7. La clave primaria de esta tabla es *name* , *lastName*, *subjectCode*, *semester* y *year*.

Una vez que el usuario ha seleccionado el o los atributos que forman parte de la clave primaria, el usuario da clic en el botón empezar y ya estamos listos para empezar a normalizar la tabla.

ASISTENTE DE NORMALIZACIÓN													
Cargar Archivo Tabla universal Primera Forma Normal Segunda Forma Normal Tercera Forma Normal													
name	lastName	birdDate	address	city	phone	subjectCode	subjectTitle	subjectType	room	capacity	grade	semester	year
Daniel	Boss	2000-01-12	Boulevard Los Alisos 323	Loja	0987654321, 0989090987	LOI0012	Programación de Algoritmos	Troncal	4A	25	28	Oct - Feb	2019
Jean	Vernier	1998-02-25	Bernardo Valdiviezo 123-23	Quito	NaN	AKS0013	Desarrollo Espiritual I	Formación básica	7B	24	40	Abr - Ago	2017
Juan	Macas	2002-04-16	Juan José Peña 898-01	Cuenca	0912873432	BDA0001	Bases de Datos Avanzadas	Troncal	7B	24	34	Abr - Ago	2018
Jairo	Samaniego	1999-12-01	Orillas del Zamora 12-12	Guayaquil	0989098909	OLD0909	Obtención y limpieza de datos	Troncal	12A	30	31	Oct - Feb	2020
Dario	Benitez	2002-12-12	Jacinto Vera 34-23	Zamora	0987654444, 0987097867, 2222098	ESF6789	Economía	Genérica	4A	25	39	Oct - Feb	2017
Stephan	Pasteur	2002-04-11	Los boulevares 323	Machala	2345 908	LOI0012	Programación de Algoritmos	Troncal	5C	28	34	Abr - Ago	2015
Daniel	Chardin	1999-02-01	Av N 25 12-09	Salinas	0989098789	BDA0001	Bases de Datos Avanzadas	Troncal	12A	30	35	Oct - Feb	2014

Fig. 6. Tabla universal desnormalizada.

Seleccione los atributos que forman parte de la Clave primaria

- ☒ name
- ☒ lastName
- ☐ birdDate
- ☐ address
- ☐ city
- ☐ phone
- ☒ subjectCode
- ☐ subjectTitle
- ☐ subjectType
- ☐ room
- ☐ capacity
- ☐ grade
- ☒ semester
- ☒ year

EMPEZAR

Fig. 7. Formulario para ingresar clave primaria

Seguidamente se muestra al usuario la tabla universal, Figura 8, y se pide que seleccione los atributos no clave que contienen datos no atómicos y el separador de los datos no atómicos, ver Figura 9. En nuestro ejemplo, el campo *phone* contiene datos no atómicos.

ASISTENTE DE NORMALIZACIÓN													
Cargar Archivo Tabla universal Primera Forma Normal Segunda Forma Normal Tercera Forma Normal													
Su Clave primaria es ['name', 'lastName', 'subjectCode', 'semester', 'year']													
name	lastName	birdDate	address	city	phone	subjectCode	subjectTitle	subjectType	room	capacity	grade	semester	year
Daniel	Boss	2000-01-12	Boulevard Los Alisos 323	Loja	0987654321, 0989090987	LOI0012	Programación de Algoritmos	Troncal	4A	25	28	Oct - Feb	2019
Jean	Vernier	1998-02-25	Bernardo Valdiviezo 123-23	Quito	NaN	AKS0013	Desarrollo Espiritual I	Formación básica	7B	24	40	Abr - Ago	2017
Juan	Macas	2002-04-16	Juan José Peña 898-01	Cuenca	0912873432	BDA0001	Bases de Datos Avanzadas	Troncal	7B	24	34	Abr - Ago	2018
Jairo	Samaniego	1999-12-01	Orillas del Zamora 12-12	Guayaquil	0989098909	OLD0909	Obtención y limpieza de datos	Troncal	12A	30	31	Oct - Feb	2020
Dario	Benitez	2002-12-12	Jacinto Vera 34-23	Zamora	0987654444, 0987097867, 2222098	ESF6789	Economía	Genérica	4A	25	39	Oct - Feb	2017
Stephan	Pasteur	2002-04-11	Los boulevares 323	Machala	2345 908	LOI0012	Programación de Algoritmos	Troncal	5C	28	34	Abr - Ago	2015
Daniel	Chardin	1999-02-01	Av N 25 12-09	Salinas	0989098789	BDA0001	Bases de Datos Avanzadas	Troncal	12A	30	35	Oct - Feb	2014

Fig. 8. Tabla universal y su clave primaria

Seleccione los atributos no clave que contienen datos no atómicos

- ☐ birdDate
- ☐ address
- ☐ city
- ☒ phone
- ☐ subjectTitle
- ☐ subjectType
- ☐ room
- ☐ capacity
- ☐ grade

Seleccione el separador de los datos no atómicos

Coma

PRIMERA FORMA NORMAL

Fig. 9. Formulario para seleccionar datos no atómicos

Al presionar el botón PRIMERA FORMA NORMAL, se procede a realizar la primera forma normal, que consiste en eliminar grupos de datos repetidos y celdas con datos no atómicos, ver Figura 10.

PRIMERA FORMA NORMAL													
Su Clave primaria es ['name', 'lastName', 'subjectCode', 'semester', 'year']													
name	lastName	birdDate	address	city	subjectCode	subjectTitle	subjectType	room	capacity	grade	semester	year	
Daniel	Boss	2000-01-12	Boulevard Los Alisos 323	Loja	LOI0012	Programación de Algoritmos	Troncal	4A	25	28	Oct - Feb	2019	
Jean	Vernier	1998-02-25	Bernardo Valdiviezo 123-23	Quito	AKS0013	Desarrollo Espiritual I	Formación básica	7B	24	40	Abr - Ago	2017	
Juan	Macas	2002-04-16	Juan José Peña 898-01	Cuenca	BDA0001	Bases de Datos Avanzadas	Troncal	7B	24	34	Abr - Ago	2018	
Jairo	Samaniego	1999-12-01	Orillas del Zamora 12-12	Guayaquil	OLD0909	Obtención y limpieza de datos	Troncal	12A	30	31	Oct - Feb	2020	
Dario	Benitez	2002-12-12	Jacinto Vera 34-23	Zamora	ESF6789	Economía	Genérica	4A	25	39	Oct - Feb	2017	
Stephan	Pasteur	2002-04-11	Los boulevares 323	Machala	LOI0012	Programación de Algoritmos	Troncal	5C	28	34	Abr - Ago	2015	
Daniel	Chardin	1999-02-01	Av N 25 12-09	Salinas	BDA0001	Bases de Datos Avanzadas	Troncal	12A	30	35	Oct - Feb	2014	

name	lastName	subjectCode	semester	year	phone
Juan	Macas	BDA0001	Abr - Ago	2018	0912873432
Jairo	Samaniego	OLD0909	Oct - Feb	2020	0989098909
Stephan	Pasteur	LOI0012	Abr - Ago	2015	2345 908
Daniel	Chardin	BDA0001	Oct - Feb	2014	0989098789
Daniel	Boss	LOI0012	Oct - Feb	2019	0987654321
Daniel	Boss	LOI0012	Oct - Feb	2019	0989090987
Dario	Benitez	ESF6789	Oct - Feb	2017	0987654444
Dario	Benitez	ESF6789	Oct - Feb	2017	0987097867
Dario	Benitez	ESF6789	Oct - Feb	2017	2222098

Fig. 10. Tablas en Primera Forma Normal

Estando en primera forma normal, podemos continuar hacia la segunda forma normal. Entonces, el sistema pregunta al usuario si existe dependencia funcional parcial, ver Figura 11. Si el usuario responde que NO, puede dar clic en continuar, la o las tablas ya se encuentran en segunda forma normal. Si responde SI debe seleccionar el subconjunto de atributos de la clave primaria de la que dependen los atributos no clave y los atributos que dependen de este subconjunto. Finalmente, el usuario da clic en continuar y pasa a segunda forma normal.

Para nuestro ejemplo, *subjectCode*, que es parte de la clave primaria, determina funcionalmente a *subjectTitle*.

¿Existen atributos no clave que dependan de forma funcional y parcial de la Clave Primaria?

SI

Seleccione el subconjunto de atributos de la clave primaria de la que dependen los atributos

- ☐ name
☐ lastName
☒ subjectCode
☐ semester
☐ year

Seleccione los atributos que dependen funcionalmente del subconjunto de atributos de la clave primaria

- ☐ birdDate
☐ address
☐ city
☒ subjectTitle
☐ subjectType
☐ room
☐ capacity
☐ grade

CONTINUAR

Fig. 11. Formulario para seleccionar dependencia funcional parcial

El sistema devuelve al usuario las tablas en segunda forma normal, ver Figura 12.

SEGUNDA FORMA NORMAL												
Su Clave primaria es ['name', 'lastName', 'subjectCode', 'semester', 'year']												
name	lastName	birdDate	address	city	subjectCode	subjectType	room	capacity	grade	semester	year	
Daniel	Boss	2000-01-12	Boulevard Los Alisos 323	Loja	LOI0012	Troncal	4A	25	28	Oct - Feb	2019	
Jean	Vernier	1998-02-25	Bernardo Valdiviezo 123-23	Quito	AKS0013	Formación básica	7B	24	40	Abr - Ago	2017	
Juan	Macas	2002-04-16	Juan José Peña 898-01	Cuenca	BDA0001	Troncal	7B	24	34	Abr - Ago	2018	
Jairo	Samaniego	1999-12-01	Orillas del Zamora 12-12	Guayaquil	OLD0909	Troncal	12A	30	31	Oct - Feb	2020	
Dario	Benitez	2002-12-12	Jacinto Vera 34-23	Zamora	ESF6789	Genérica	4A	25	39	Oct - Feb	2017	
Stephan	Pasteur	2002-04-11	Los boulevares 323	Machala	LOI0012	Troncal	5C	28	34	Abr - Ago	2015	
Daniel	Chardin	1999-02-01	Av N 25 12-09	Salinas	BDA0001	Troncal	12A	30	35	Oct - Feb	2014	

name	lastName	subjectCode	semester	year	phone
Juan	Macas	BDA0001	Abr - Ago	2018	0912873432
Jairo	Samaniego	OLD0909	Oct - Feb	2020	0989098909
Stephan	Pasteur	LOI0012	Abr - Ago	2015	2345 908
Daniel	Chardin	BDA0001	Oct - Feb	2014	0989098789
Daniel	Boss	LOI0012	Oct - Feb	2019	0987654321
Daniel	Boss	LOI0012	Oct - Feb	2019	0989890987
Dario	Benitez	ESF6789	Oct - Feb	2017	0987654444
Dario	Benitez	ESF6789	Oct - Feb	2017	0987097867
Dario	Benitez	ESF6789	Oct - Feb	2017	2222098

subjectCode	subjectTitle
LOI0012	Programación de Algoritmos
AKS0013	Desarrollo Espiritual I
BDA0001	Bases de Datos Avanzadas
OLD0909	Obtención y limpieza de datos
ESF6789	Economía
LOI0012	Programación de Algoritmos
BDA0001	Bases de Datos Avanzadas

Fig. 12. Tablas en Segunda Forma Normal

Finalmente, estando ya en segunda forma normal, podemos continuar hacia la tercera forma normal. Entonces, preguntamos al usuario si existe dependencia funcional transitiva, ver Figura 13. Si el usuario que NO, puede dar clic en 'continuar'

y ya está en tercera forma normal. En cambio, si responde que SI, debe seleccionar el o los atributos no clave que determinan de forma funcional transitiva a otros atributos no clave y cuáles son estos. En nuestro ejemplo, hay dependencia funcional transitiva desde *room* hacia *capacity*.

¿Existen atributos no clave que dependan funcionalmente de otros atributos no clave?

SI

Seleccione los atributos determinantes

- ☐ birdDate
☐ address
☐ city
☐ subjectTitle
☐ subjectType
☒ room
☐ capacity
☐ grade

Seleccione los atributos que tienen dependencia funcional transitiva

- ☐ birdDate
☐ address
☐ city
☐ subjectTitle
☐ subjectType
☐ room
☒ capacity
☐ grade

CONTINUAR

Fig. 13. Formulario para seleccionar dependencia funcional transitiva

Por último, el usuario da clic en continuar y pasa a tercera forma normal, y se muestran las tablas normalizadas, ver Figuras 14 y 15.

VI. CONCLUSIONES Y TRABAJOS FUTUROS

La normalización de base de datos relacionales es un proceso utilizado para organizar una base de datos en tablas y columnas. La idea principal con esto es que una tabla debe ser sobre un tema específico y solo incluir atributos que describan la entidad representada.

En el contexto de este trabajo, los autores buscan contribuir con el proceso de enseñanza-aprendizaje de normalización. En el proceso implementado, las tablas resultantes se limitan a un propósito, esto reduce la cantidad de datos duplicados contenidos en su base de datos. En consecuencia se elimina algunos problemas derivados de las modificaciones de la base de datos.

Para lograr el objetivo, en este trabajo se han implementado algunas reglas establecidas. A medida que aplica estas reglas, se forman nuevas tablas. La progresión de no normalizado a optimizado pasa por varias formas normales.

Se han implementado tres formas normales a las que se adhieren la mayoría de las bases de datos. A medida que las tablas satisfacen cada formulario de normalización de la base de datos sucesiva, se vuelven menos propensas a las anomalías de modificación de la base de datos y se centran más en un único propósito o tema. Antes de continuar, asegúrese de comprender la definición de una tabla de base de datos.

ASISTENTE DE NORMALIZACIÓN										
<div>☰</div>										
TABLAS NORMALIZADAS										
Su Clave primaria es ['name', 'lastName', 'subjectCode', 'semester', 'year']										
name	lastName	birdDate	address	city	subjectCode	subjectType	room	grade	semester	year
Daniel	Boss	2000-01-12	Boulevard Los Alisos 323	Loja	LOI0012	Troncal	4A	28	Oct - Feb	2019
Jean	Vernier	1998-02-25	Bernardo Valdiviezo 123-23	Quito	AKS0013	Formación básica	7B	40	Abr - Ago	2017
Juan	Macas	2002-04-16	Juan José Peña 898-01	Cuenca	BDA0001	Troncal	7B	34	Abr - Ago	2018
Jairo	Samaniego	1999-12-01	Orillas del Zamora 12-12	Guayaquil	OLD0909	Troncal	12A	31	Oct - Feb	2020
Dario	Benitez	2002-12-12	Jacinto Vera 34-23	Zamora	ESF6789	Genérica	4A	39	Oct - Feb	2017
Stephan	Pasteur	2002-04-11	Los boulevares 323	Machala	LOI0012	Troncal	5C	34	Abr - Ago	2015
Daniel	Chardin	1999-02-01	Av N 25 12-09	Salinas	BDA0001	Troncal	12A	35	Oct - Feb	2014

Fig. 14. Tabla principal en Tercera Forma Normal

En este trabajo se presenta la herramienta diseñada y se describen algunas experimentaciones realizadas en un ambiente controlado. Se planea como trabajo futuro ampliar el alcance de validación con el fin de determinar oportunidades de mejora en la herramienta que permitan mejorar el proceso de aprendizaje de normalización. Respecto a la herramienta, como trabajo a futuro queda implementar una funcionalidad para verificar que la selección del usuario de claves primarias sea correcta. Además se planea mejorar la detección automatizada de dependencias funcionales.

Los autotres planean convertir los datos modelados a través de la heramienta en datos abiertos [8]. El propóstiso es producir recursos educativos abiertos (OER, *Open Educational Resources*) que apoyen la educación de ingenieros [9] en temas relacionados con bases de datos relacionales.

REFERENCES

- [1] Piza-Davila, Ivan Gutiérrez-Preciado, Luis Guzmán, Víctor H. (2017). An educational software for teaching database normalization. *Computer Applications in Engineering Education*. 25. 10.1002/cae.21838.
- [2] Joshi, Abhijit Mendjoge, Neha Narvekar, Meera. (2016). Intelligent Tutoring System for Database Normalization. 10.1109/IC-CUBEA.2016.7860013.
- [3] Dongare, Y. Dhabe, Priyadarshan Deshmukh, S.. (2011). RDBNorma: - A semi-automated tool for relational database schema normalization up to third normal form. *Computing Research Repository - CORR*. 3. 10.5121/ijdms.2011.3109.
- [4] Bahmani, Amir Naghibzadeh, M. Bahmani, Behnam. (2008). Automatic database normalization and primary key generation. 000011 - 000016. 10.1109/CCECE.2008.4564486.
- [5] Yazici, Ali Karakaya, Ziya. (2007). JMathNorm: A Database Normalization Tool Using Mathematica. 186-193. 10.1007/978-3-540-72586-2-27.

name	lastName	subjectCode	semester	year	phone
Juan	Macas	BDA0001	Abr - Ago	2018	0912873432
Jairo	Samaniego	OLD0909	Oct - Feb	2020	0989098909
Stephan	Pasteur	LOI0012	Abr - Ago	2015	2345 908
Daniel	Chardin	BDA0001	Oct - Feb	2014	0989098789
Daniel	Boss	LOI0012	Oct - Feb	2019	0987654321
Daniel	Boss	LOI0012	Oct - Feb	2019	0989890987
Dario	Benitez	ESF6789	Oct - Feb	2017	0987654444
Dario	Benitez	ESF6789	Oct - Feb	2017	0987097867
Dario	Benitez	ESF6789	Oct - Feb	2017	2222098

subjectCode	subjectTitle
LOI0012	Programación de Algoritmos
AKS0013	Desarrollo Espiritual I
BDA0001	Bases de Datos Avanzadas
OLD0909	Obtención y limpieza de datos
ESF6789	Economía
LOI0012	Programación de Algoritmos
BDA0001	Bases de Datos Avanzadas

room	capacity
4A	25
7B	24
7B	24
12A	30
4A	25
5C	28
12A	30

Fig. 15. Tablas en Tercera Forma Normal

- [6] Ram, Shiu. (2008). Teaching Data Normalization: Traditional Classroom Methods versus Online Visual Methods—a Literature Review.
- [7] UNAM. Normalización de Bases de Datos. Unidad de apoyo para el aprendizaje. Recuperado de https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/872/mod_resource/content/1/contenido/index.html
- [8] Tovar, E., Piedra, N. (2014). Guest editorial: open educational resources in engineering education: various perspectives opening the education of engineers. *IEEE Transactions on Education*, 57(4), 213-219.
- [9] Piedra, N., Chicaiza, J., López, J., Caro, E. T. (2017). A rating system that open-data repositories must satisfy to be considered OER: Reusing open data resources in teaching. In 2017 IEEE Global Engineering Education Conference (EDUCON) (pp. 1768-1777). IEEE.
- [10] Codd, E. F. (1979). Extending the database relational model to capture more meaning. *ACM Transactions on Database Systems (TODS)*, 4(4), 397-434.
- [11] Codd, E. F. (2002). A relational model of data for large shared data banks. In *Software pioneers* (pp. 263-294). Springer, Berlin, Heidelberg.
- [12] Ramez Elmasri and Shamkant B. Navathe. 2015. *Fundamentals of Database Systems* (7th. ed.). Pearson.
- [13] Python Software Foundation. <https://www.python.org/>
- [14] Pandas - Python Data Analysis Library. <https://pandas.pydata.org/>
- [15] Flask. <https://flask.palletsprojects.com/>
- [16] Heroku. <https://www.heroku.com/>