



**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE SISTEMAS**

**BASE DE DATOS I**

**Proyecto Final**

**Andrés Tayupanta y Felipe Vallejo**

**PROFESOR: Miguel Ortiz**

**Junio, 2023**

## **Introducción:**

“El proceso de normalización consiste en una serie de reglas que deben cumplir las tablas y relaciones obtenidas tras el paso del modelo entidad relación al modelo relacional, para entonces ser un modelo lógico” (Mendoza y López, 2018). Al aplicar la normalización, se busca eliminar redundancias de datos, salvaguardar la integridad de la información y prevenir problemas en las operaciones de inserción, modificación, eliminación y consulta. Estas reglas, al ser aplicadas correctamente, conducen a una base de datos más estructurada y eficiente, lo cual simplifica su mantenimiento y mejora la calidad de los datos almacenados.

El modelo combinatorio puede aplicarse en el diseño de una base de datos relacional al considerar las combinaciones posibles de entidades y atributos. Por ejemplo, al diseñar las tablas de una Base de Datos, es necesario decidir qué atributos se incluirán en cada tabla y cómo se relacionarán entre sí. Aquí aparece la lógica combinatoria para determinar todas las combinaciones posibles y elegir la estructura más adecuada para diseñar la estructura de las tablas, determinar las combinaciones de entidades y atributos, y realizar operaciones y consultas.

En el contexto del proyecto de investigación llevado a cabo por el Departamento de Informática de la Universidad XYZ, la aplicación de la normalización resulta fundamental para la gestión eficiente de la información relacionada con proyectos, profesores, publicaciones y líneas de investigación. La base de datos diseñada para este propósito debe cumplir con las reglas de normalización, de manera que garantice la consistencia y la integridad de los datos almacenados, así como la optimización de las consultas y operaciones realizadas sobre la base de datos.

Por otro lado, es importante conocer el concepto de dependencias funcionales para aplicar las reglas de normalización. “Una dependencia funcional (DF) es una restricción que se establece entre dos conjuntos de atributos de la base de datos.” (Quiroz, 2013). El propósito de las dependencias funcionales es identificar los problemas de redundancia y anomalías de actualización en la base de datos.

## **Problemática:**

**Complejidad de las relaciones:** El modelo de datos presenta múltiples relaciones entre las entidades, lo cual puede resultar en una estructura compleja y difícil de gestionar. Es importante asegurarse de que las relaciones estén correctamente definidas y que se apliquen las restricciones adecuadas para mantener la integridad de los datos.

**Asignación de supervisores:** Existe una relación de supervisión entre profesores doctores y no doctores. La asignación de supervisores puede ser complicada, ya que un profesor no doctor puede tener un único supervisor en un momento dado. Es importante establecer mecanismos para gestionar y mantener actualizada esta información.

**Gestión de la temporalidad:** El proyecto involucra el manejo de fechas de inicio y finalización tanto para los proyectos como para la participación de los profesores. La gestión de la temporalidad requiere un diseño cuidadoso para asegurar la consistencia de los datos y la adecuada representación de los periodos de tiempo.

**Publicaciones y líneas de investigación:** El proyecto involucra la relación entre publicaciones y líneas de investigación. Una publicación puede abarcar varias líneas de investigación, y un profesor puede escribir sobre una línea de investigación en una publicación. La gestión de estas relaciones puede ser compleja y requerir una estructura adecuada para garantizar la precisión y coherencia de los datos.

### **Objetivos:**

#### **General:**

- Aplicar el proceso de normalización en el modelo de datos del Departamento de Informática de la Universidad XYZ, con el fin de crear un modelo eficiente y no redundante. Esto implicará estructurar la información relacionada con los proyectos de investigación, profesores, publicaciones y líneas de investigación de manera coherente y sin repeticiones innecesarias.

#### **Específicos:**

- Identificar y analizar las dependencias funcionales presentes en el modelo de datos existente del Departamento de Informática de la Universidad XYZ, tomando en cuenta las características y supuestos semánticos descritos en el texto.
- Establecer las restricciones necesarias en el modelo de datos para asegurar la consistencia y validez de las relaciones entre las entidades, evitando inserciones, actualizaciones o eliminaciones que puedan violar las dependencias funcionales establecidas.
- Aplicar las formas normales para diseñar el modelo de datos con el objetivo de asegurar que todas las entidades y atributos estén correctamente estructurados, no existan valores múltiples o compuestos y eliminar las dependencias funcionales parciales y transitivas, respectivamente.

### **Justificación:**

Este trabajo está enfocado en explicar las distintas formas normales a través de un caso práctico, con la finalidad de demostrar la importancia de la normalización en bases de datos. Para ello, se analizará el texto y se aplicarán las reglas de normalización con el objetivo de crear un modelo sin redundancias y eficiente.

Además, este trabajo busca abordar los desafíos específicos presentes en el ámbito académico de un departamento universitario, como la gestión eficiente de proyectos de investigación, la colaboración entre profesores, la publicación de resultados y el seguimiento de líneas de investigación. Estos aspectos son de vital importancia para el Departamento de Informática de la Universidad XYZ, ya que contribuyen al avance del conocimiento y al desarrollo de la investigación en el campo de la informática.

Al aplicar las formas normales en el diseño de la base de datos, se logrará una estructura bien organizada y optimizada, eliminando redundancias y anomalías en los datos. Esto permitirá obtener beneficios significativos, como la reducción del espacio de almacenamiento necesario, una mejor gestión de la información y una mayor precisión en los resultados obtenidos.

Además, la normalización de la base de datos garantiza la integridad de los datos, evitando la inserción de información inconsistente o contradictoria. Al establecer relaciones claras entre las entidades, se facilita la consulta y recuperación de información, lo que contribuye a una gestión más eficiente de los proyectos de investigación y las publicaciones generadas.

### **Desarrollo:**

#### **ENUNCIADO**

El Departamento de Informática de la Universidad XYZ necesita una base de datos para almacenar la información concerniente a los proyectos de investigación tanto actuales como pasados en los que trabajan los profesores y así poder llevar a cabo una gestión más eficiente. La información que se desea almacenar corresponde a los siguientes supuestos semánticos. En el departamento los profesores participan en proyectos de investigación caracterizados por un código de referencia único, por un nombre, un acrónimo, un presupuesto total, el programa de I+D que lo financia, una fecha de inicio y una fecha de finalización y una breve descripción de los objetivos del proyecto. En los proyectos trabajan profesores del departamento durante un periodo de tiempo, es decir, una fecha de inicio y una fecha de fin, pudiendo ocurrir que un profesor trabaje en el mismo proyecto en varias épocas ( $f_{ini}$ ,  $f_{fin}$ ) diferentes. Un profesor se identifica por su nombre y apellidos y se caracteriza por su despacho y teléfono y puede trabajar en varios proyectos simultáneamente y en un proyecto de investigación trabajan varios profesores. De todos los profesores que trabajan en el proyecto hay uno que es el investigador principal de proyecto que interesa conocer. Es importante tener en cuenta que el profesor investigador del proyecto nunca varía a lo largo de la vida del proyecto de investigación. Los profesores pueden ser doctores o no doctores, de tal manera que un profesor no doctor siempre tiene a un único profesor doctor como supervisor en un momento determinado, interesando almacenar los supervisores y periodos de tiempo de la supervisión que ha tenido un determinado profesor no doctor. En relación con la participación de los profesores en proyectos de investigación, el investigador principal de un proyecto siempre tiene que ser un doctor. Por otro lado, los proyectos de investigación producen una serie de publicaciones sobre las que también interesa guardar información. Una publicación se caracteriza por un número en secuencia dentro de cada proyecto de investigación y se guardará el título y los profesores que la han escrito; las publicaciones son de dos tipos, publicaciones en congresos y publicaciones en revista; de las primeras se almacenará el nombre del congreso, su tipo (nacional o internacional), la fecha de inicio y de fin, el lugar de celebración, país y la editorial que ha publicado las actas del congreso (si es que se han publicado); de las publicaciones en revista interesa saber el nombre de la revista, la editorial, el volumen, el número y las páginas de inicio y fin. No solamente interesa conocer los profesores que han participado en las publicaciones de los proyectos de investigación sino también las líneas de investigación que cubren estas publicaciones. Una línea de

investigación se identifica por un código, un nombre (por ejemplo, “recuperación de información multilingüe”, “bases de datos espacio-temporales”, etc.) y un conjunto de descriptores (por ejemplo, la línea de investigación “bases de datos temporales” puede tener como descriptores “Bases de Datos”, “SGBD Relacional”, “Dimensión temporal”). Los profesores tendrán asociados en la BD las líneas de investigación en las que trabajan incluso podría ocurrir que hubiera profesores que no tuvieran ninguna línea asignada. Así, tanto los profesores doctores como los no doctores pueden escribir publicaciones sobre una o más líneas de investigación y nos interesa saber sobre qué línea de investigación ha escrito un determinado profesor en una publicación, teniendo en cuenta que un profesor que participa en una publicación solo escribe en el ámbito de una línea de investigación y que una determinada publicación puede cubrir varias líneas de investigación. Por último, aparte de la información especificada para los proyectos de investigación también se almacenarán las líneas de investigación que abarca cada proyecto.

### **Análisis:**

- Con base en el enunciado, se ha determinado que los profesores participan en varios proyectos de investigación y un proyecto de investigación incluye a varios profesores. Cada proyecto tiene su propia fecha de inicio y fin.
- Además, el texto menciona que un profesor puede tener varias participaciones en un proyecto de investigación en épocas diferentes. Por ello, se infiere que el profesor y la investigación determinan las fechas de inicio y fin de la participación. De forma que las fechas juegan un rol importante dentro del proceso.
- Se menciona que existe un investigador principal por proyecto, el cual tiene que ser un doctor y no varía a lo largo del tiempo, es decir, está solamente a cargo de un proyecto y no puede ser transferido o modificado. Así que en este caso hemos identificado que el profesor y la investigación determinan si un profesor es líder o no, teniendo en cuenta las restricciones de negocio.
- Después se menciona que los profesores no doctores tienen a un solo supervisor doctor en un momento determinado. De esta parte se infirió que la supervisión era de la participación del proyecto en un momento determinado, y que el supervisor era la misma persona que el líder de investigación.
- Por otro lado, los proyectos de investigación producen varias publicaciones que pueden ser de revista o de congreso, cada una con características diferentes.
- Las publicaciones incluyen varias líneas de investigación, las cuales tienen un conjunto de descriptores. En este caso, hemos podido evidenciar que la línea de investigación determina a los descriptores, debido a que hacen referencia a las palabras clave dentro de las líneas de investigación.
- Después, se menciona que se interesa almacenar las publicaciones y líneas de investigación en las que trabaja un profesor y las líneas de investigación que abarca cada proyecto.

## Dependencias funcionales

### Entidad PROFESOR:

**prf\_id (pk)** -> prf\_nombres, prf\_apellidos, prf\_num\_despacho, prf\_telefono, prf\_titulacion

### Entidad PROFESOR\_INVESTIGACION (Intersección):

**prf\_id, inv\_id** -> prf\_inv\_fecha\_inicio, prf\_inv\_fecha\_fin, prf\_inv\_es\_lider

### Entidad INVESTIGACION:

**inv\_id (pk)** -> inv\_nombre, inv\_acronimo, inv\_descripcion, inv\_financiador, inv\_presupuesto, inv\_fecha\_inicio, inv\_fecha\_fin

### Entidad PUBLICACION:

**pub (pk)** -> pub\_titulo

### Entidad REVISTA:

**rev\_id (pk)** -> rev\_nombre, rev\_editorial, rev\_volumen, rev\_numero, rev\_pag\_inicio, rev\_pag\_fin

### Entidad CONGRESO:

**con\_id (pk)** -> con\_nombre, con\_tipo, con\_fecha\_inicio, con\_fecha\_fin, con\_lugar\_celebracion, con\_pais, con\_editorial

### Entidad LINEA\_INVESTIGACION:

**lninv\_id (pk)** -> lninv\_nombre, lninv\_descriptores

Al identificar las dependencias funcionales, se puede observar que todos están correctamente normalizados.

### Tablas:

Tabla	Descripción
PROFESOR	Tabla que almacena la información de los profesores del departamento informático de la Universidad
PROFESOR_INVESTIGACION	Tabla que almacena la información de los profesores que participan en proyectos de investigación
INVESTIGACION	Tabla que almacena información de los proyectos de investigación
PUBLICACION	Tabla que almacena la información de la publicación
REVISTA	Tabla que almacena la información de las revistas
CONGRESO	Tabla que almacena la información de los congresos
LINEA_INVESTIGACION	Tabla que almacena la información de las líneas de publicación

### Atributos:

ENTIDADES	ATRIBUTOS	DESCRIPCION	Tipo de dato
PROFESOR	prf_id (pk)	Campo que contiene el identificador único para cada profesor	varchar
	prf_nombres	Campo que contiene los nombres del profesor	varchar
	prf_apellidos	Campo que contiene los apellidos del profesor	varchar
	prf_num_despacho	Campo que contiene el número de despacho del profesor	integer
	prf_telefono	Campo que contiene el número de telefono del profesor	varchar
	prf_titulacion	Campo que contiene la titulación del profesor. Ej: Doctor, Magister, etc	varchar
PROFESOR_INVESTIGACION	prfinv_fecha_ini	Campo que contiene la fecha de inicio de la participación de un profesor en un proyecto	date
	prfinv_fecha_fin	Campo que contiene la fecha de fin de la participación de un profesor en un proyecto	date
	prfinv_es_lider	Campo que verifica si un profesor es lider de investigación	number (1,0)
INVESTIGACION	inv_id (pk)	Campo que contiene un código de referencia único para un proyecto de investigación	varchar
	inv_nombre	Campo que contiene el nombre del proyecto de investigación	varchar
	inv_acronimo	Campo que contiene el acrónimo del proyecto de investigación	varchar
	inv_descripcion	Campo que contiene la descripción del proyecto de investigación	varchar
	inv_financiador	Campo que contiene el programa de I+D que financia el proyecto de investigación	varchar
	inv_presupuesto	Campo que contiene el presupuesto del proyecto de investigación	number(12,4)
	inv_fecha_inicio	Campo que contiene la fecha de inicio del proyecto de investigación	date
	inv_fecha_fin	Campo que contiene la fecha de fin del proyecto de investigación	date
	pub_id (pk)	Campo que contiene el identificador único para cada publicación	integer
	pub_titulo	Campo que contiene el título para cada publicación	varchar
REVISTA	rev_id (pk)	Campo que contiene el identificador único para cada revista	varchar
	rev_nombre	Campo que contiene el nombre de revista	varchar
	rev_editorial	Campo que contiene la editorial con la que la revista fue publicada	varchar
	rev_volumen	Campo que contiene el volumen de la revista	integer
	rev_numero	Campo que contiene el número de la revista	integer
	rev_pag_inicio	Campo que contiene la página de inicio de la revista	integer
	rev_pag_fin	Campo que contiene la página de fin de la revista	integer
CONGRESO	con_id (pk)	Campo que contiene el identificador único para cada congreso	varchar
	con_nombre	Campo que contiene el nombre del congreso	varchar
	con_tipo	Campo que contiene el tipo del congreso	varchar
	con_fecha_ini	Campo que contiene la fecha de inicio del congreso	date
	con_fecha_fin	Campo que contiene la fecha de fin del congreso	date
	con_lugar_celebracion	Campo que contiene el lugar de celebración del congreso	varchar
	con_pais	Campo que contiene el país del congreso	varchar
	con_editorial	Campo que contiene la editorial del con la que se ha publicado las actas de congreso	varchar
LINEA_INVESTIGACION	lininv_id (pk)	Campo que contiene el identificador único para cada línea de investigación	varchar
	lininv_nombre	Campo que contiene el nombre de la línea de investigación	varchar
	lininv_descriptores	Campo que contiene los descriptores de la línea de investigación. Ej "Bases de Datos", "SGBD Relacional", "Dimension temporal", etc	varchar

## Modelo Combinatorio:

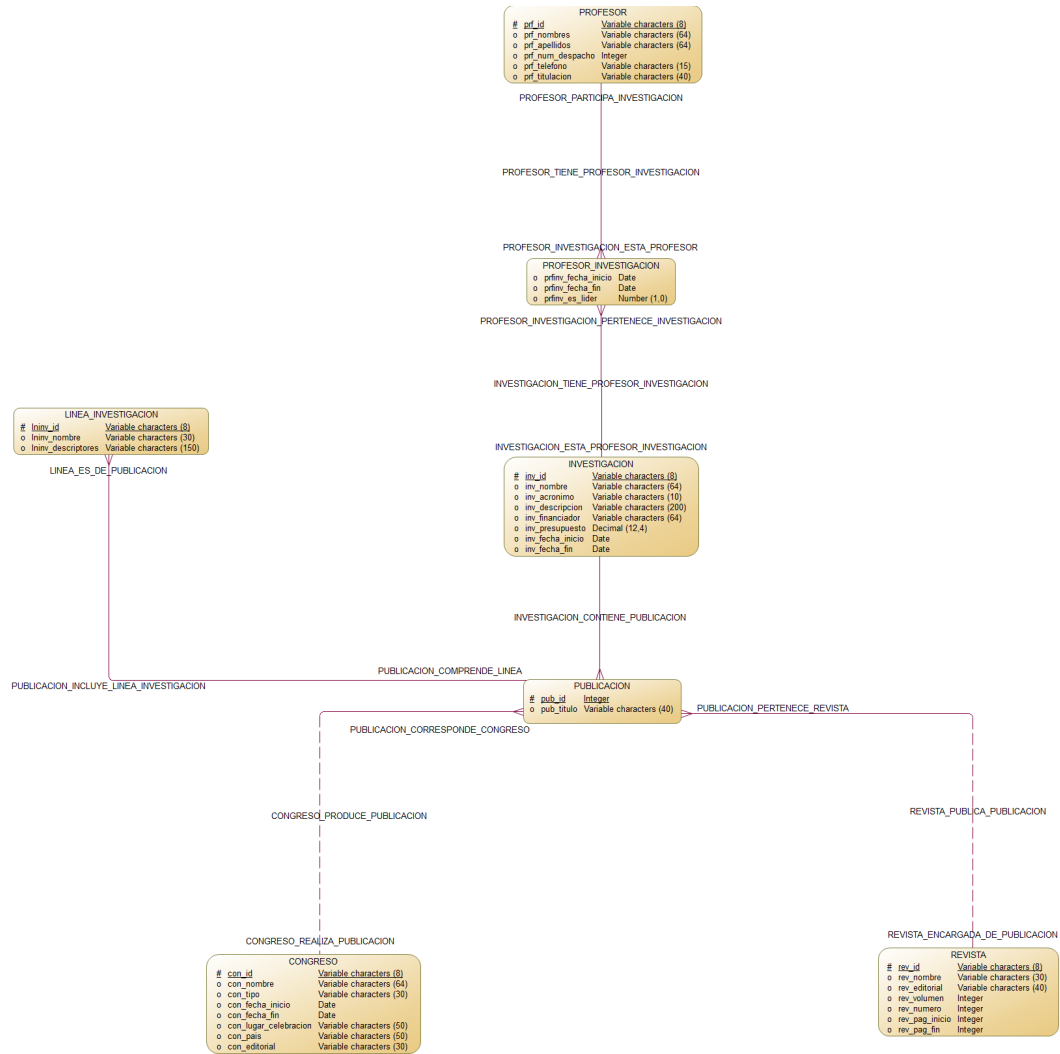
PROFESOR	PROFESOR_INVESTIGACION	1:N
	INVESTIGACIÓN	...
	PUBLICACION	...
	REVISTA	...
	CONGRESO	...
	LINEA_INVESTIGACION	...
PROFESOR_INVESTIGACION	INVESTIGACIÓN	N:1
	PUBLICACION	...
	REVISTA	...
	CONGRESO	...
	LINEA_INVESTIGACION	...

INVESTIGACION	PUBLICACION	1:N
	REVISTA	...
	CONGRESO	...
	LINEA_INVESTIGACION	...
	DESCRIPTOR	...
	DESCRIPTOR_LINEA	...
PUBLICACION	REVISTA	N:1
	CONGRESO	N:1
	LINEA_INVESTIGACION	1:N
REVISTA	CONGRESO	...
	LINEA_INVESTIGACION	...
CONGRESO	LINEA_INVESTIGACION	...

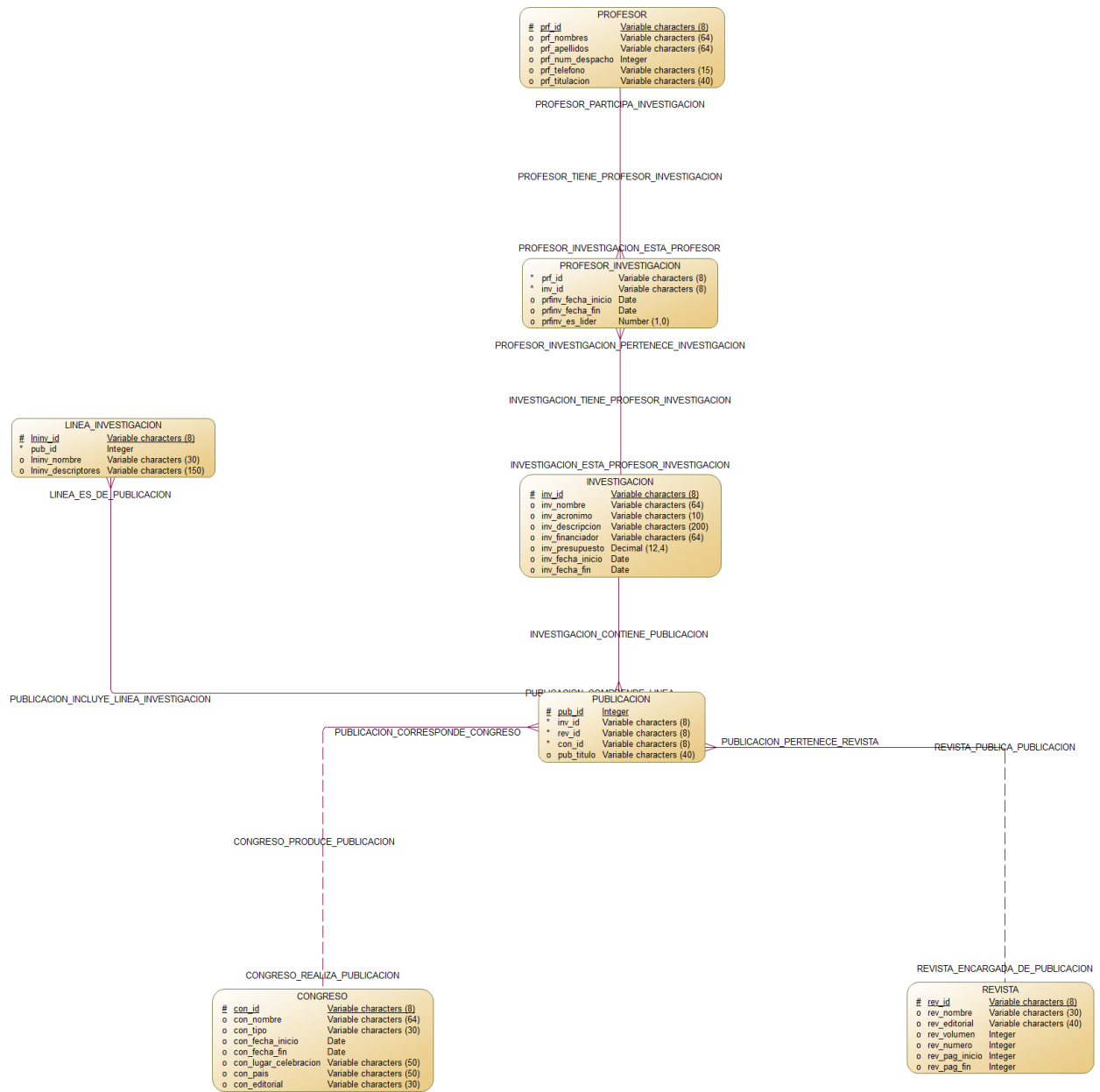
**DISEÑO:**

**Modelo conceptual:**

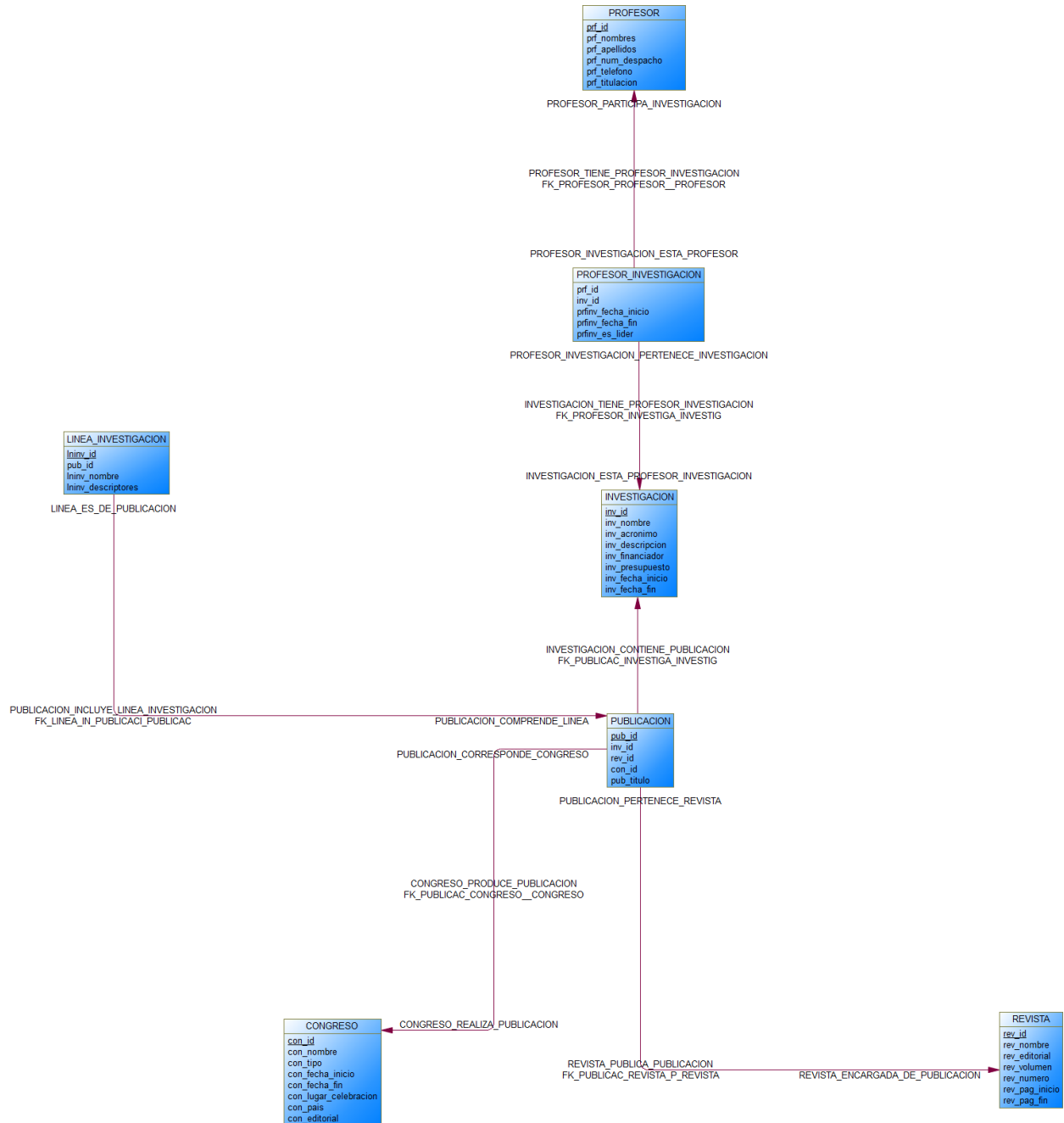




**Modelo Lógico:**



**Modelo Físico:**



### Asignación de espacio de almacenamiento para los tablespace DATOSG1 e INDICESG1:

En primer lugar, se ha decidido obtener la tabla del modelo que más atributos tiene y que más espacio ocupa, con la finalidad de realizar un análisis sobre la memoria a utilizar al implementar la base de datos.

INVESTIGACION		
#	<u>inv_id</u>	<u>Variable characters (8)</u>
o	inv_nombre	Variable characters (64)
o	inv_acronimo	Variable characters (10)
o	inv_descripcion	Variable characters (200)
o	inv_financiador	Variable characters (64)
o	inv_presupuesto	Decimal (12,4)
o	inv_fecha_inicio	Date
o	inv_fecha_fin	Date

- inv\_id -> 8 caracteres o bytes
- inv\_nombre -> 64 caracteres o bytes
- inv\_acronimo -> 10 caracteres o bytes
- inv\_descripcion -> 200 caracteres o bytes
- inv\_financiador -> 64 caracteres o bytes
- inv\_presupuesto -> 8 bytes
- inv\_fecha\_inicio -> 7 bytes
- inv\_fecha\_fin -> 7 bytes

Se desea crear 15 proyectos de investigación, así que partiendo de esa base se realiza el primer cálculo:  $(15) * (8+64+10+200+64+8+7+7) = 5520$  bytes o 0.00527573 MB

Sabemos que los profesores participan en varios proyectos de investigación, y que un proyecto incluye a varios profesores, por lo tanto es necesario tener una cantidad establecida de profesores y participaciones. Se insertarán 4 profesores en cada proyecto de investigación ( $4*15$ ) con el objetivo de validar las reglas de negocio dentro de la base de datos.

PROFESOR		
#	<u>prf_id</u>	<u>Variable characters (8)</u>
o	prf_nombres	Variable characters (64)
o	prf_apellidos	Variable characters (64)
o	prf_num_despacho	Integer
o	prf_telefono	Variable characters (15)
o	prf_titulacion	Variable characters (40)

- prf\_id -> 8 caracteres o bytes
- prf\_nombres -> 64 caracteres o bytes
- prf\_apellidos -> 64 caracteres o bytes

- prf\_num\_despacho -> 4 bytes
- prf\_telefono -> 15 caracteres o bytes
- prf\_titulacion-> 40 caracteres o bytes

$$(4*15) * (8+64+64+4+15+40) = 11700 \text{ bytes o } 0.01114 \text{ MB}$$

Todos los profesores participarán en un proyecto de investigación, por lo tanto se realizará el cálculo con base en las mismas condiciones que el anterior:

PROFESOR_INVESTIGACION		
o	prfinv_fecha_inicio	Date
o	prfinv_fecha_fin	Date
o	prfinv_es_lider	Number

- prfinv\_fecha\_inicio -> 7 bytes
- prfinv\_fecha\_fin -> 7 bytes
- prfinv\_es\_lider -> 2 bytes

$$(4*15) * (7+7+2) = 960 \text{ bytes o } 0.00091553 \text{ MB}$$

Después, se necesitarán al menos 5 publicaciones de revista y 5 publicaciones de congreso por cada investigación. Para ello se ingresaran 7 congresos y 8 revistas que estarán asociados a las publicaciones de cada investigación:

CONGRESO		
#	con_id	Variable characters (8)
o	con_nombre	Variable characters (64)
o	con_tipo	Variable characters (30)
o	con_fecha_inicio	Date
o	con_fecha_fin	Date
o	con_lugar_celebracion	Variable characters (50)
o	con_pais	Variable characters (50)
o	con_editorial	Variable characters (30)

- con\_id -> 8 caracteres o bytes
- con\_nombre -> 64 caracteres o bytes
- con\_tipo -> 30 caracteres o bytes
- con\_fecha\_inicio -> 7 bytes
- con\_fecha\_fin -> 7 bytes
- con\_lugar\_celebracion -> 50 caracteres o bytes
- con\_pais-> 50 caracteres o bytes
- con\_editorial -> 30 caracteres o bytes

$$(7) * (8+64+30+7+7+50+50+30) = 1722 \text{ bytes o } 0.001642 \text{ MB}$$

REVISTA		
#	rev_id	Variable characters (8)
o	rev_nombre	Variable characters (30)
o	rev_editorial	Variable characters (40)
o	rev_volumen	Integer
o	rev_numero	Integer
o	rev_pag_inicio	Integer
o	rev_pag_fin	Integer

- rev\_id -> 8 caracteres o bytes
- rev\_nombre -> 30 caracteres o bytes
- rev\_editorial -> 40 caracteres o bytes
- rev\_volumen -> 4 bytes
- rev\_numero -> 4 bytes
- rev\_pag\_inicio -> 4 bytes
- rev\_pag\_fin -> 4 bytes

$$(8) * (8+30+40+4+4+4+4) = 752 \text{ bytes o } 0.000716 \text{ MB}$$

Teniendo estos valores, se procederá a obtener el total de espacio que ocuparán las 10 publicaciones de cada investigación (15).

PUBLICACION		
#	pub_id	Integer
o	pub_titulo	Variable characters (40)

- pub\_id -> 4 bytes
- pub\_titulo -> 40 caracteres o bytes

$$(10*15) * (4+40) = 66000 \text{ bytes o } 0.06293 \text{ MB}$$

Se tomará una publicación de cada investigación y se le asignarán 5 líneas de investigación a cada una (15\*5)

LINEA_INVESTIGACION		
#	linv_id	Variable characters (8)
o	linv_nombre	Variable characters (10)
o	linv_descriptores	Variable characters (150)

- linv\_id -> 8 bytes o caracteres
- linv\_nombre -> 10 bytes o caracteres
- linv\_descriptores -> 150 bytes o caracteres.

$$(15*5) * (8+10+150) = 12600 \text{ bytes o } 0.01202 \text{ MB}$$

TOTAL:  $0.00527573 + 0.01114 + 0.00091553 + 0.001642 + 0.000716 + 0.06293 + 0.01202$   
= 0.09428126 MB ( aproximadamente 1 MB para satisfacer las necesidades actuales)

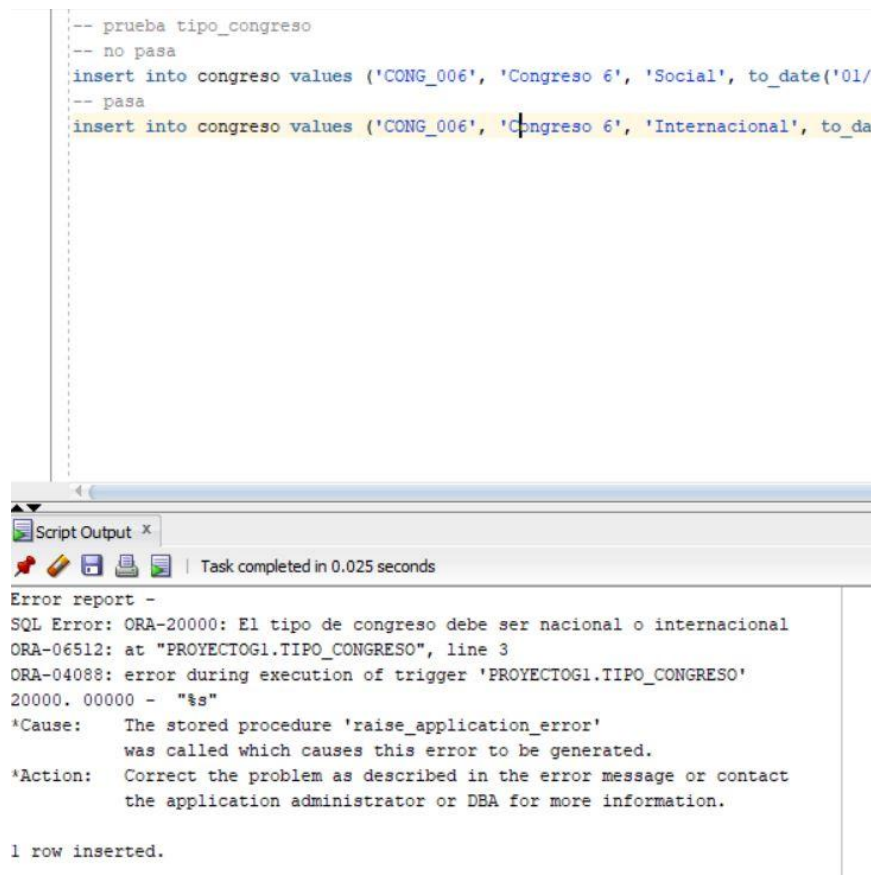
Finalmente, hemos decidido duplicar el tamaño para evitar problemas en caso de requerir más registros y para el autoextend hemos decidido utilizar la mitad del tamaño. Mientras que para el tablespace INDICESG1 hemos decidido utilizar el mismo tamaño.

TOTAL = 2 MB

AUTOEXTEND = 1MB

## IMPLEMENTACIÓN Y PRUEBAS:

A continuación haremos una breve demostración de los triggers que hemos creado,



```
-- prueba tipo_congreso
-- no pasa
insert into congreso values ('CONG_006', 'Congreso 6', 'Social', to_date('01/
-- pasa
insert into congreso values ('CONG_006', 'Congreso 6', 'Internacional', to_da
```

Script Output x

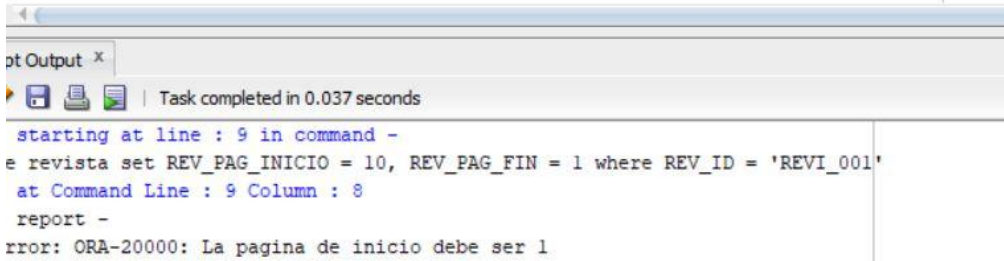
Task completed in 0.025 seconds

Error report -  
SQL Error: ORA-20000: El tipo de congreso debe ser nacional o internacional  
ORA-06512: at "PROYECTOGL.TIPO\_CONGRESO", line 3  
ORA-04088: error during execution of trigger 'PROYECTOGL.TIPO\_CONGRESO'  
20000. 00000 - "%s"  
\*Cause: The stored procedure 'raise\_application\_error'  
was called which causes this error to be generated.  
\*Action: Correct the problem as described in the error message or contact  
the application administrator or DBA for more information.

1 row inserted.

Creamos un trigger **tipo\_congreso** que su función principal es asegurarse de que en cada registro el parámetro que pasamos en tipo de congreso únicamente pueda ser nacional o internacional, aquí se muestra que al poner “social” nos muestra un mensaje de error.

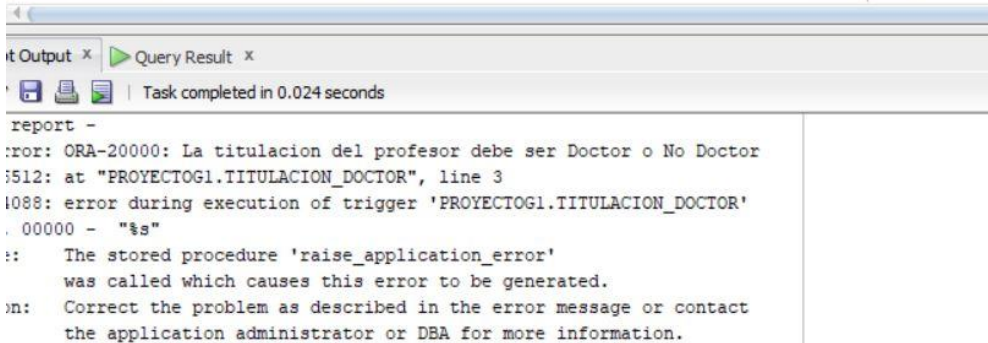
```
-- prueba paginas_revista
-- no pasa
update revista set REV_PAG_INICIO = 10, REV_PAG_FIN = 1 where REV_ID = 'REVI_001';
-- pasa
update revista set REV_PAG_INICIO = 1, REV_PAG_FIN = 6 where REV_ID = 'REVI_001';
```



The screenshot shows the SQL Developer interface. The top pane contains the SQL code. The bottom pane, titled "Output", shows the execution results. The first update statement failed with the error: "ORA-20000: La pagina de inicio debe ser 1". The second update statement executed successfully.

Aquí creamos un trigger **paginas\_revista** el cual se asegura de que la página de inicio de la revista siempre sea 1.

```
-- prueba titulacion_doctor
select * from profesor;
-- no pasa
insert into profesor values ('PROF_006', 'Luis', 'Gonzales', 6, '123456', 'Ingeniero');
-- pasa
insert into profesor values ('PROF_006', 'Luis', 'Gonzales', 6, '123456', 'Doctor');
```



The screenshot shows the SQL Developer interface. The top pane contains the SQL code. The bottom pane, titled "Output", shows the execution results. The first insert statement failed with the error: "ORA-20000: La titulacion del profesor debe ser Doctor o No Doctor". The second insert statement executed successfully.

Aquí creamos un trigger **titulacion\_doctor** que únicamente permite que se ingrese “doctor” o “no doctor” en la titulación del profesor.



```
-- prueba lider_investigacion
-- no pasa
insert into profesor_investigacion values ('PROF_002', 'INVE_001', to_date('01/01/2015', 'dd/mm/yyyy'), to_date('01/01/2016', 'dd/mm/yyyy'), 1);
-- pasa
insert into profesor_investigacion values ('PROF_003', 'INVE_001', to_date('01/01/2015', 'dd/mm/yyyy'), to_date('01/01/2016', 'dd/mm/yyyy'), 0);
```

Output x Query Result x

Task completed in 0.04 seconds

```
report -
Error: ORA-20000: Ya existe un lider en la investigacion
512: at "PROYECTOGL.LIDER_INVESTIGACION", line 6
088: error during execution of trigger 'PROYECTOGL.LIDER_INVESTIGACION'
00000 - "$s"
Cause: The stored procedure 'raise_application_error'
was called which causes this error to be generated.
Action: Correct the problem as described in the error message or contact
the application administrator or DBA for more information.

inserted.
```

Aquí hicimos un trigger **lider\_investigacion** que se asegura de no permitir que se agregue otro líder en la investigación en el caso de que ya exista uno.

```
-- prueba fecha_investigacion
-- no pasa
insert into investigacion values ('INVE_006', 'Investigacion 6', 'INV_6', 'Descripcion 6', 'Financiador 6', 6000, to_date('01/01/2020', 'dd/mm/yyyy'), to_date('01/01/2019', 'dd/mm/yyyy'));
-- pasa
insert into investigacion values ('INVE_007', 'Investigacion 7', 'INV_7', 'Descripcion 7', 'Financiador 7', 7000, to_date('01/01/2018', 'dd/mm/yyyy'), to_date('01/01/2019', 'dd/mm/yyyy'));
```

Script Output x Query Result x

Task completed in 0.025 seconds

```
Error report -
SQL Error: ORA-20000: La fecha de inicio debe ser menor que la fecha de fin
ORA-06512: at "PROYECTOGL.FECHA_INVESTIGACION", line 3
ORA-04088: error during execution of trigger 'PROYECTOGL.FECHA_INVESTIGACION'
20000. 00000 - "$s"
Cause: The stored procedure 'raise_application_error'
was called which causes this error to be generated.
Action: Correct the problem as described in the error message or contact
the application administrator or DBA for more information.

1 row inserted.
```

El trigger **fecha\_investigacion** lo que se encarga de hacer es validar lo siguiente: investigación tiene dos atributos, fecha de inicio y fecha de fin, se debe validar que la fecha de inicio sea menor que la fecha de fin.

```
-- prueba lider_doctor
-- no pasa
insert into profesor_investigacion values ('PROF_005', 'INVE_002', to_date('01/01/2016', 'dd/mm/yyyy'), to_date('01/01/2017', 'dd/mm/yyyy'), 1);
-- pasa
update profesor_investigacion set PRFINV_ES_LIDER = 1 where PRF_ID = 'PROF_002';
```

Script Output x Query Result x

Task completed in 0.039 seconds

```
insert into profesor_investigacion values ('PROF_005', 'INVE_002', to_date('01/01/2016', 'dd/mm/yyyy'), to_date('01/01/2017', 'dd/mm/yyyy'), 1)
Error at Command Line : 34 Column : 13
Error report -
SQL Error: ORA-20000: El profesor debe ser doctor para ser lider
```

El trigger **lider\_doctor** se asegura de que únicamente permita que prfinv\_es\_lider en profesor\_investigacion sea 1 cuando el profesor sea doctor en prf\_titulacion.

```
-- prueba fecha_profesor_investigacion
-- no pasa
insert into profesor_investigacion values ('PROF_002', 'INVE_005', to_date('01/01/2020', 'dd/mm/yyyy'), to_date('01/01/2019', 'dd/mm/yyyy'), 0);
-- pasa
insert into profesor_investigacion values ('PROF_002', 'INVE_005', to_date('01/01/2019', 'dd/mm/yyyy'), to_date('01/01/2020', 'dd/mm/yyyy'), 0);
```

Script Output x Query Result x

Task completed in 0.028 seconds

20000. 00000 - "s"

\*Cause: The stored procedure 'raise\_application\_error' was called which causes this error to be generated.

\*Action: Correct the problem as described in the error message or contact the application administrator or DBA for more information.

1 row inserted.

El trigger **fecha\_profesor\_investigacion** se asegura de que la fecha de inicio sea siempre menor a la fecha de fin en la tabla profesor\_investigacion.

```
-- prueba fecha_fin_profesor_investigacion
-- no pasa
insert into profesor_investigacion values ('PROF_001', 'INVE_003', to_date('01/01/2017', 'dd/mm/yyyy'), to_date('01/01/2020', 'dd/mm/yyyy'), 0);
-- pasa
insert into profesor_investigacion values ('PROF_001', 'INVE_003', to_date('01/01/2017', 'dd/mm/yyyy'), to_date('01/01/2018', 'dd/mm/yyyy'), 0);
```

Script Output x Query Result x

Task completed in 0.027 seconds

SQL Error: ORA-20000: La fecha de fin de la investigacion que participa el profesor no puede ser mayor a la fecha de fin de la investigacion

ORA-06512: at "PROYECTO1.FECHA\_FIN\_PROFESOR\_INVESTIGACION", line 6

ORA-04088: error during execution of trigger 'PROYECTO1.FECHA\_FIN\_PROFESOR\_INVESTIGACION'

20000. 00000 - "s"

\*Cause: The stored procedure 'raise\_application\_error' was called which causes this error to be generated.

\*Action: Correct the problem as described in the error message or contact the application administrator or DBA for more information.

1 row inserted.

El trigger **fecha\_fin\_profesor\_investigacion** se asegura de que la fecha de fin de profesor\_investigacion no sea mayor a la de fin en investigación que se pasa en el segundo parámetro.

```
-- prueba fecha_inicio_profesor_investigacion
-- no pasa
insert into profesor_investigacion values ('PROF_001', 'INVE_002', to_date('01/01/2015', 'dd/mm/yyyy'), to_date('01/01/2016', 'dd/mm/yyyy'), 0);
-- pasa
insert into profesor_investigacion values ('PROF_001', 'INVE_002', to_date('01/01/2016', 'dd/mm/yyyy'), to_date('01/01/2017', 'dd/mm/yyyy'), 0);
```

Script Output x Query Result x

Task completed in 0.032 seconds

Error at Command Line : 52 Column : 13

Error report -

SQL Error: ORA-20000: La fecha de inicio de la investigacion que participa el profesor no puede ser menor a la fecha de inicio de la investigacion

ORA-06512: at "PROYECTO1.FECHA\_INICIO\_PROFESOR\_INVESTIGACION", line 6

ORA-04088: error during execution of trigger 'PROYECTO1.FECHA\_INICIO\_PROFESOR\_INVESTIGACION'

20000. 00000 - "s"

\*Cause: The stored procedure 'raise\_application\_error' was called which causes this error to be generated.

\*Action: Correct the problem as described in the error message or contact the application administrator or DBA for more information.

El trigger **fecha\_inicio\_profesor\_investigacion** se asegura de que la fecha de inicio de la investigación sea menor a la fecha de inicio pasada en profesor\_investigacion.

```
-- prueba pagina_inicio_fin_revista
-- no pasa
insert into revista values ('REVI_006', 'Revista 6', 'Editorial 6', 6, 6, 60, 60);
-- pasa
insert into revista values ('REVI_006', 'Revista 6', 'Editorial 6', 6, 6, 1, 61);
```

Script Output x Query Result x

Task completed in 0.028 seconds

Error report -  
SQL Error: ORA-20000: La pagina de inicio y fin no pueden ser iguales  
ORA-06512: at "PROYECTO1.PAGINA\_INICIO\_FIN\_REVISTA", line 3  
ORA-04088: error during execution of trigger 'PROYECTO1.PAGINA\_INICIO\_FIN\_REVISTA'  
20000. 00000 - "%s"  
\*Cause: The stored procedure 'raise\_application\_error'  
was called which causes this error to be generated.  
\*Action: Correct the problem as described in the error message or contact  
the application administrator or DBA for more information.

**pagina\_inicio\_fin\_revista** es para que la página de inicio y fin no sean igual en revista, rev\_pagino\_inicio, rev\_pagina\_fin.

```
-- prueba participacion_profesor_investigacion
-- no pasa
insert into profesor_investigacion values ('PROF_001', 'INVE_001', to_date('01/01/2015', 'dd/mm/yyyy'), to_date('01/01/2016', 'dd/mm/yyyy'), 0);
```

Script Output x Query Result x

Task completed in 0.065 seconds

1 row inserted.

Error starting at line : 64 in command -  
insert into profesor\_investigacion values ('PROF\_001', 'INVE\_001', to\_date('01/01/2015', 'dd/mm/yyyy'), to\_date('01/01/2016', 'dd/mm/yyyy'), 0)  
Error at Command Line : 64 Column : 13  
Error report -  
SQL Error: ORA-20001: Ya participo en esas fechas  
ORA-06512: at "PROYECTO1.PARTICIPACION\_PROFESOR\_INVESTIGACION", line 12  
ORA-04088: error during execution of trigger 'PROYECTO1.PARTICIPACION\_PROFESOR\_INVESTIGACION'

**participacion\_profesor\_investigacion** es un trigger que no permita que vuelva a participar en las fechas que ya participó un profesor.

```
-- prueba trg_validar_publicacion
-- no pasa
insert into publicacion values (6, 'INVE_006', 'REVI_006', 'CONG_003', 'Titulo 6');
-- pasa
insert into publicacion values (7, 'INVE_007', null, 'CONG_004', 'Titulo 7');|
```

Script Output x Query Result x

Task completed in 0.026 seconds

Error starting at line : 68 in command -  
insert into publicacion values (6, 'INVE\_006', 'REVI\_006', 'CONG\_003', 'Titulo 6')

Error at Command Line : 68 Column : 13

Error report -  
SQL Error: ORA-20001: Una publicación debe ser asociada a un congreso o a una revista, pero no a ambos o ninguno.  
ORA-06512: at "PROYECTOGL.TRG\_VALIDAR\_PUBLICACION", line 9  
ORA-04088: error during execution of trigger 'PROYECTOGL.TRG\_VALIDAR\_PUBLICACION'

1 row inserted.

**trg\_validar\_publicacion** válida que una publicación debe ser asociada a un congreso o a una revista, pero no a ambos o ninguno.

```
-- prueba trg_verificar_participacion_lider
-- no pasa
insert into profesor_investigacion values ('PROF_004', 'INVE_003', to_date('01/01/2017', 'dd/mm/yyyy'), to_date('03/01/2017', 'dd/mm/yyyy'), 1);
-- pasa
insert into profesor_investigacion values ('PROF_004', 'INVE_003', to_date('01/01/2017', 'dd/mm/yyyy'), to_date('01/01/2018', 'dd/mm/yyyy'), 1);
```

Script Output x Query Result x

Task completed in 0.039 seconds

Error starting at line : 74 in command -  
insert into profesor\_investigacion values ('PROF\_004', 'INVE\_003', to\_date('01/01/2017', 'dd/mm/yyyy'), to\_date('03/01/2017', 'dd/mm/yyyy'), 1)

Error at Command Line : 74 Column : 13

Error report -  
SQL Error: ORA-20003: Las fechas del lider de investigación deben coincidir con las fechas de la investigación  
ORA-06512: at "PROYECTOGL.TRG\_VERIFICAR\_PARTICIPACION\_LIDER", line 12  
ORA-04088: error during execution of trigger 'PROYECTOGL.TRG\_VERIFICAR\_PARTICIPACION\_LIDER'

1 row inserted.

**trg\_verificar\_participacion\_lider** se asegura de que las fechas del líder de investigación deben coincidir con las fechas de la investigación.



```
-- prueba prf_num_despacho y prf_telefono
-- no pasa
insert into profesor values ('PROF_006', 'Luis', 'Gonzales', 5, '123456', 'No Doctor');
insert into profesor values ('PROF_006', 'Luis', 'Gonzales', 6, '12345', 'No Doctor');
-- pasa
insert into profesor values ('PROF_006', 'Luis', 'Gonzales', 6, '123456', 'No Doctor');
```

Script Output x Query Result x

Task completed in 0.026 seconds

Error starting at line : 80 in command -  
insert into profesor values ('PROF\_006', 'Luis', 'Gonzales', 5, '123456', 'No Doctor')  
Error at Command Line : 80 Column : 13  
Error report -  
SQL Error: ORA-20001: Ya existe un profesor con ese numero de despacho  
ORA-06512: at "PROYECTO1.PRF\_NUM\_DESPACHO", line 9  
ORA-04088: error during execution of trigger 'PROYECTO1.PRF\_NUM\_DESPACHO'

Error starting at line : 81 in command -  
insert into profesor values ('PROF\_006', 'Luis', 'Gonzales', 6, '12345', 'No Doctor')  
Error at Command Line : 81 Column : 13  
Error report -  
SQL Error: ORA-20001: Ya existe un profesor con ese numero de telefono  
ORA-06512: at "PROYECTO1.PRF\_TELEFONO", line 9  
ORA-04088: error during execution of trigger 'PROYECTO1.PRF\_TELEFONO'

**prf\_num\_despacho y prf\_telefono** se aseguran de que no se puedan insertar registros en la tabla profesor donde, ya se repita el número de despacho y el número de teléfono.

### **Conclusiones:**

- Con base en el trabajo realizado, se ha podido determinar que las formas normales proporcionan criterios para evaluar y mejorar la estructura de la base de datos, eliminando redundancias y asegurando la integridad de los datos.
- Se ha evidenciado que aplicar las reglas de negocio en la base de datos permite garantizar la consistencia e integridad de los datos de acuerdo con las necesidades y requisitos del negocio.
- Se ha comprobado que las dependencias funcionales permiten identificar y eliminar redundancias en los datos.

### **Recomendaciones:**

- Se recomienda comprender las reglas de normalización para poder identificar las formas normales y las reglas que las definen.
- Para normalizar, se recomienda realizar un análisis detallado de las tablas y establecer de forma precisa cuáles son las dependencias funcionales existentes entre los atributos.
- Se recomienda realizar la documentación para ayudar a los usuarios finales a entender cómo se estructura la base de datos, qué información se almacena en ella y cómo se relacionan las diferentes tablas y entidades.

**Bibliografía:**

- Mendoza A., A., & López, R. (2018). Objetivos Diseño de Base de datos. *Repositorio Academico de La Universidad de Chile*, 1–9.  
<https://repositorio.uchile.cl/handle/2250/151632>
- Quiroz, J. (2013). El modelo relacional de bases de datos. *Boletín de Política Informática*, 6, 53–61.