



# Modelamiento y Normalización de datos

Parte II





# Inicio

**{desafío}**  
latam\_



15 Minutos

*/\*Reconocer la utilidad del diccionario de datos.\*/*

*/\*Elaborar un diccionario de datos detallando un modelo relacional.\*/*

## Objetivo

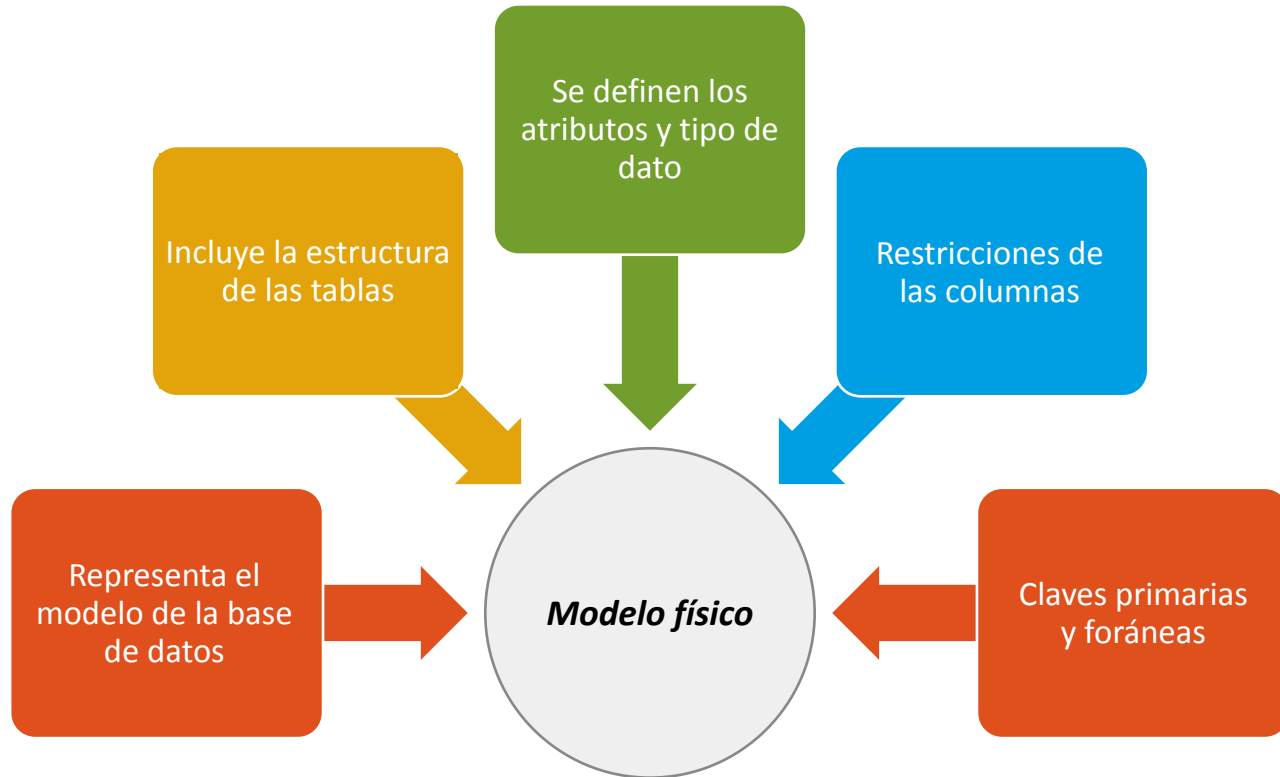


# Desarrollo

{desafío}  
latam\_

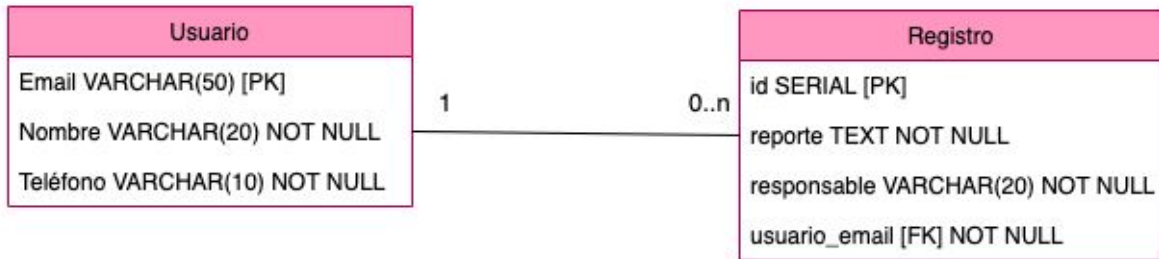


150 Minutos



# Traspassando el modelo físico a SQL

Una compañía telefónica designa un departamento para recibir llamadas de los usuarios que presenten problemas con la señal en sus teléfonos o la plataforma online, y necesita almacenar todos los reportes como registros en la base de datos. Tenemos el siguiente modelo:



# Traspassando el modelo físico a SQL

En el siguiente código verás el ejemplo de cómo se construirían las tablas con consultas SQL:

```
CREATE TABLE usuario(  
  email VARCHAR(50),  
  nombre VARCHAR(20) NOT NULL,  
  telefono VARCHAR(15) NOT NULL,  
  PRIMARY KEY (email)  
);  
CREATE TABLE registro(  
  id NUMBER,  
  reporte TEXT NOT NULL,  
  responsable VARCHAR(50) NOT NULL,  
  usuario_email VARCHAR(50) REFERENCES usuario  
  (email),  
  PRIMARY KEY (id)  
);
```



# Traspassando el modelo físico a SQL

- Existen programas que nos permiten generar SQL a partir de modelos físicos y otros que nos permiten generar los diagramas a partir de SQL.
- Ejecuta el código dentro de una base de datos y agregar los siguientes datos para probar el modelo:

```
INSERT INTO usuario(email, nombre, telefono) VALUES
('usuario1@gmail.com',
 'Juan', '12345678');
INSERT INTO usuario(email, nombre, telefono)
VALUES ('usuario2@gmail.com',
 'Francisca', '12345679');
INSERT INTO registro (reporte, responsable,
usuario_email) values ('El usuario
presenta problemas para realizar el pago online',
'Javiera',
'usuario1@gmail.com');
INSERT INTO registro (reporte, responsable,
usuario_email) values ('El usuario
presenta problemas para ingresar a la plataforma',
'Javiera',
'usuario2@gmail.com');
```

## Ahora te toca a ti

Crear las tablas correspondientes al modelo físico de la siguiente imagen, la cual corresponde a una plataforma que ofrece reseña de películas. Este modelo tiende a tener registros duplicados y redundancia de datos, es decir, no está normalizado pero lo estaremos normalizando en próximos ejercicios propuestos.



## Resolvamos en conjunto

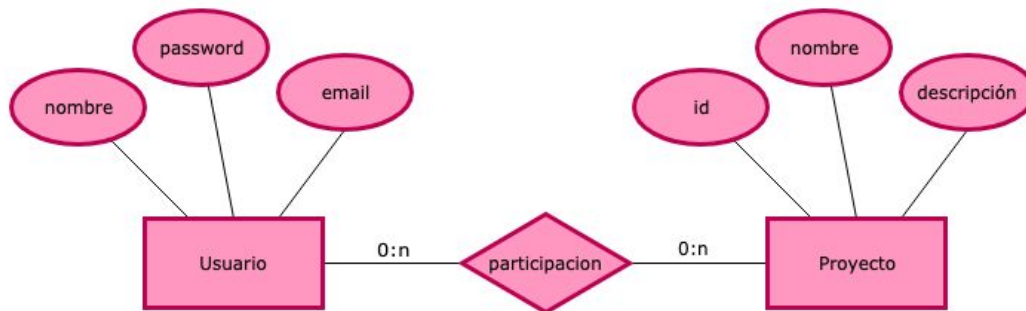
Un empleado puede trabajar en diversos proyectos de una empresa, los empleados para entrar a la plataforma necesitan identificarse con un correo corporativo, password y su nombre. De cada proyecto se tiene el nombre y una descripción.

Se pueden dar situaciones donde en un proyecto no trabaje ningún empleado y un empleado no trabaje en ningún proyecto. Procedamos con la modelación de este caso:

**Caso con  
relaciones N a  
N**

# Modelo conceptual

1. Identificar entidades
2. Agrupar entidades con sus atributos
3. Identificar relaciones y sus cardinalidades
4. Identificar candidatos únicos



Continuando con el caso propuesto de la plataforma de reseñas de películas, se debe realizar el diagrama del diseño conceptual, considerando la relación entre ambas entidades.

**Ahora te toca a  
ti**

# Modelo lógico

1. Transformar todas las entidades en tablas y agregar los atributos como columnas de la tabla.
2. Transformar todas las relaciones del tipo N:N en tablas.
3. Propagar la clave primaria de las tablas en las relaciones 1:N desde el lado de las 1 al lado de las N.

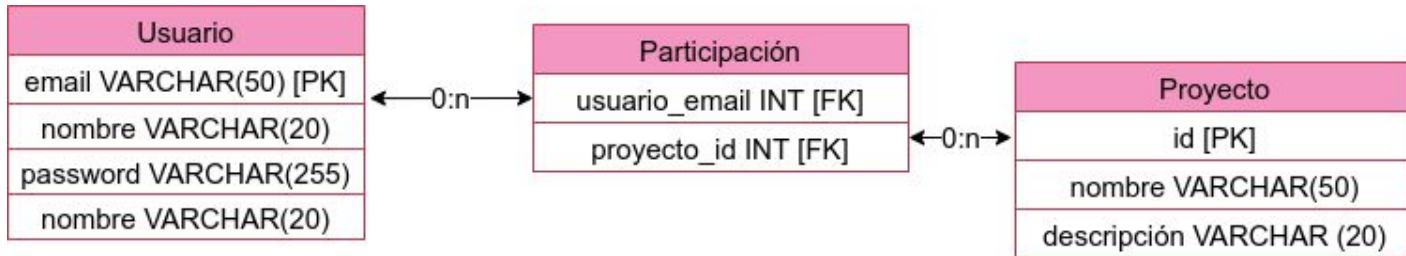


Continuando con el caso propuesto de la plataforma de reseñas de películas, se debe diseñar el modelo lógico basado en el resultado del ejercicio propuesto 2.

**Ahora te toca a  
ti**

# Modelo físico

Una vez teniendo el modelo lógico, el único cambio que debemos hacer es incluir en los atributos el tipo de dato y la longitud de este para pasarlo a un modelo físico, pues este será el último dato necesario para pasar a la construcción de las tablas.





Continuando con el caso propuesto de la plataforma de reseñas de películas, se debe diseñar el modelo físico basado en el resultado del ejercicio propuesto 3.

**Ahora te toca a  
ti**

# Normalización

# ¿Para qué sirve la normalización?

Evitar la  
redundancia de  
datos.



Simplificar la  
actualización de  
datos.



Garantizar la  
integridad  
referencial  
(prevenir  
inconsistencia en  
las relaciones).

# Notación

- Utilizaremos la siguiente notación para detallar la normalización de una tabla:

```
nombre_tabla(atributo_1, atributo_2, ..., atributo_n)
```

- Fuera de los paréntesis se tiene el nombre de la tabla con la que se trabajará.
- Dentro de los paréntesis se encuentran los atributos existentes.
- Hay casos en que dentro de una tabla pueden existir atributos repetidos. Para definirlos de forma explícita, se deben encapsular entre llaves { } de la siguiente forma:

```
nombre_tabla(atributo_1, atributo_2, ..., atributo_n,  
{atributo_repetido})
```

## ¿Qué es un grupo repetitivo?

Lo más fácil es verlo con un ejemplo, tenemos la siguiente tabla Cliente:

N Cliente	Nombre	Teléfonos
1	Fernanda	123-456 123-457
2	Felipe	234-412
3	Francisco	123-411

En este caso, puede haber más de un teléfono asociado. Por eso es un atributo (o grupo de atributos) repetitivo, lo anterior es muy similar a esta nueva tabla:

N_Cliente	Nombre	Teléfono1	Teléfono2
1	Fernanda	123-456	123-457
2	Felipe	234-412	
3	Francisco	123-411	

# Identificando las claves primarias y foráneas

- Clave primaria con el símbolo # y letras en negrita.

```
Tabla1(#clavePrimaria, atributo_1, ..., atributo_n)
```

- Claves primarias compuestas con letra negrita y signo #.

```
Tabla3(#atributoPrimario, claveForánea, #otroAtributoPrimarioCompuesto, ..., atributo_n)
```

- Clave foránea con letras cursivas.

```
Tabla2(#clavePrimaria, claveForánea, ..., atributo_n)
```

Continuando con el caso propuesto de la plataforma de reseñas de películas, escribir las tablas en la notación aprendida.

**Ahora te toca a  
ti**

# Implementación de formas normales

Para ejemplificar el uso de las formas normales, utilizaremos el siguiente ejemplo, que corresponde a la factura de un paciente en un hospital, donde se muestran los datos del paciente y los ítems que consumió.

*Resolvamos el ejemplo en conjunto:*

Factura		# 23464
		20/04/2019
Nombre paciente: Marta Fuentes		#Paciente: 4724
Dirección: Monjitas 245		Comuna: Santiago
COD-SISTEMA-SALUD: 10		Ciudad: Santiago
ITEM	NOMBRE	VALOR
200	Pieza semiprivada	150
204	Televisión	10
245	Rayos X	25
414	Exámenes	35
SUBTOTAL		220
%IMPUESTO		22
TOTAL		242



# Primera Forma Normal (1FN)

La tabla debe cumplir las siguientes condiciones:

- Cada campo o atributo deben ser atómicos, es decir debe contener un único valor.
- No pueden haber grupos repetitivos.

## Resultado:

Factura

# Factura	Fecha factura	Nombre	#Paciente	Dirección	Comuna	Ciudad	CSS
23464	20/04/19	Marta Fuentes	4724	Monjtas 245	Santiago	Santiago	10

FacturaItem

# Factura	#Item	Nombre	Valor
23464	200	Pieza semiprivada	150
23464	204	Televisión	10
23464	245	Rayos X	25
23464	414	Exámenes	35

## Ahora te toca a ti

Realizar la primera forma normal con la siguiente tabla:

Películas				
Id_Pelicula	Pelicula	Genero	ID_Actor	Actor
1	Interestelar	Ficción	1	Matthew McConaughey
1	Interestelar	Ficción	2	Anne Hathaway
2	En busca de la felicidad	Drama	3	Will Smith
2	En busca de la felicidad	Drama	4	Jaden Smith

## Segunda Forma Normal (2FN)

Esta forma debe cumplir las siguientes condiciones:

- Debe satisfacer la 1FN.
- Cada atributo debe depender de la clave primaria, y no solo una parte de ella.
- Los atributos que dependen de manera parcial de la clave primaria deben ser eliminados o almacenados en una nueva entidad.

### Resultado:

#### Paciente

#Paciente	Nombre	#Dirección	Comuna	Ciudad	CSS
4724	Marta Fuentes	Monjitas 245	Santiago	Santiago	10

#### Factura

#Factura	Fecha factura	#Paciente
23464	20/04/19	4724

Realizar la segunda forma normal con el resultado del ejercicio de las películas expuesto en el ejercicio propuesto 6.

A vertical line separates the white left side from the orange right side. Along this line, there are several large, stylized symbols: a closing curly brace '}', an '@' symbol, an opening curly brace '{', and a smiley face ':D'.

**Ahora te toca a  
ti**

## Tercera Forma Normal (3FN)

Esta forma debe cumplir las siguientes condiciones:

- Debe satisfacer 2FN.
- Toda entidad debe depender directamente de la clave primaria.

**Resultado:**

Paciente

#Paciente	Nombre	Dirección	id_comuna	CSS
4724	Marta Fuentes	Monjitas 245	145	10

Comuna

id_comuna	Nombre	id_ciudad
145	Santiago	12

Ciudad

id_ciudad	Ciudad
12	Santiago

Continuando con el ejercicio de las películas, realiza la tercera forma normal con el resultado del ejercicio propuesto 7.

**Ahora te toca a  
ti**

# Resumen de la normalización



# Desnormalización y sus usos

- Es el proceso de añadir redundancia en las tablas de manera deliberada.
- Una de las motivaciones para desnormalizar tablas en una base de datos es el reducir el tiempo de consulta al implementar joins en múltiples tablas.
- Es un paso posterior a la normalización, donde se busca maximizar la eficiencia y representación de los datos, a expensas de hacer más compleja la mantención de una tabla específica.



## Resolvamos en conjunto

Tenemos una base de datos sobre pacientes en un hospital, la cual está compuesta por tres tablas:

- Una tabla Paciente, que registra el nombre del paciente, el tipo de tratamiento y el doctor con el que se atendió, éstas últimas dos columnas están registradas como claves foráneas.

Paciente	Tratamiento_id	Doctor_id
María Zenteno	1	1
Fernando Sánchez	2	2
Jose Artigas	1	2
Emilia Tapia	1	1
Felipe Zamorano	2	1
César Oyarce	2	1
Catalina Maillard	1	2

- Una tabla Doctores que vincula la clave primaria con el nombre del doctor tratante.

Id	Nombre
1	Astorquiza
2	Feris

- Una tabla Tratamiento que vincula la clave primaria con el nombre del tratamiento asignado al paciente.

Id	Nombre
1	Control Médico
2	Biopsia

## Ahora te toca a ti

Una tienda de remates tiene vendedores que manejan diferentes almacenes en diferentes direcciones y ciudades. Tienen una base de datos normalizada, sin embargo, consideran que las consultas están tardando mucho porque son consultas muy procesadas con diferentes comandos y subqueries, por lo que se ven en la necesidad de desnormalizar las siguientes 3 tablas.

Dirección		
id	Nombre	ciudad
1	Maipu 727	1
2	Santo Domingo 1450	2
3	Santa Isabel 517	1
4	San Pablo 1361	2

Almacen			
id	Responsable	Ciudad	Dirección
1	Valentina	1	1
2	Valentina	2	2
2	Omar	1	3
2	Omar	2	4

Ciudad	
id	Nombre
1	Concepción
2	Santiago

# Diccionario de datos

- Es un repositorio de metadatos que contiene las definiciones de los objetos de datos, descripciones y relaciones entre sí.
- incluye las características lógicas y específicas de los datos, como el nombre, descripción, alias, contenido y dominio.

## Diccionario de datos

## ¿Para qué sirve el diccionario de datos?

- Permite que los analistas conozcan los detalles y descripciones de los elementos de la base de datos, sus relaciones, los permisos y usuarios asociados, obteniendo una lista de objetos que forman parte del flujo de datos de todo el sistema.

# Elaboración de un diccionario de datos

Oracle pone a disposición una serie de vistas, que nos proporcionan valiosa información del diccionario de datos.

```
-- Seleccionar las tablas y sus comentarios, si es que los tiene
```

```
SELECT table_name TABLA,  
       comments    COMENTARIO  
FROM   user_tab_comments  
WHERE  table_name NOT LIKE 'BIN$%'  
ORDER BY tabla;
```

```
-- resultado
```

TABLA	COMENTARIO
-----	-----
PARTICIPACION	
PROYECTO	
USUARIO	

Generar una sentencia SQL que extraiga la información de tus tablas para elaborar un diccionario de datos. Usa cualquiera de tus bases de datos y nombra tus atributos solo con letras.

**Ahora te toca a  
ti**





# Cierre

{desafío}  
latam\_



15 Minutos

**¿Existe algún concepto que  
no hayas comprendido?**

**Volvamos a revisar los conceptos que más te  
hayan costado antes de seguir adelante**

**Reflexionemos**





*Academia de  
talentos digitales*

[www.desafiolatam.com](http://www.desafiolatam.com)



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam