

Guía de ejercicios - Modelos Entidad-Relación y Relacional (II)



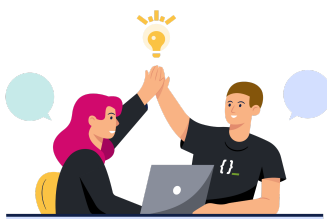
¡Hola! Te damos la bienvenida a esta nueva guía de estudio.

¿En qué consiste esta guía?

La siguiente guía de estudio tiene como objetivo practicar y ejercitar los contenidos que hemos visto en clase.

Tabla de contenidos

Actividad guiada: Aplicando modelado y cardinalidad en modelos de datos	2
Actividad guiada: Integrando normalización en un modelo de datos	4
Tabla Estudiantes	4
Análisis de normalización	5
Apliquemos la 1FN a nuestro modelo de datos	5
Entidad Cursos	5
Entidad Estudiantes	6
Dependencia Funcional	6
Dependencia Transitiva	6
Apliquemos la 2FN a nuestro modelo de datos	7
Tabla intermedia	7
Apliquemos la 3FN a nuestro modelo de datos	7
Entidad Carreras	7
Entidad modificada de Estudiantes	8
Reflexión	8
¡Manos a la obra! - Agregando cardinalidad entre Consulta y Paciente	9
Solución ejercicio propuesto	10



¡Comencemos!



Actividad guiada: Aplicando modelado y cardinalidad en modelos de datos

A continuación, aplicaremos el proceso de modelamiento de datos utilizando los conceptos aprendidos hasta el momento.

Una clínica requiere que generemos el modelo de su lógica de negocio de acuerdo a los siguiente requerimientos.

Para realizar el diagrama se recomienda utilizar <http://draw.io>.

1. Un médico tiene una especialidad.

- **Paso 1:** Generamos entonces las entidades médico y especialidad, asumiendo que estas dos tienen una relación 1 - N, es decir, un médico puede tener una especialidad y una especialidad la pueden tener varios médicos.

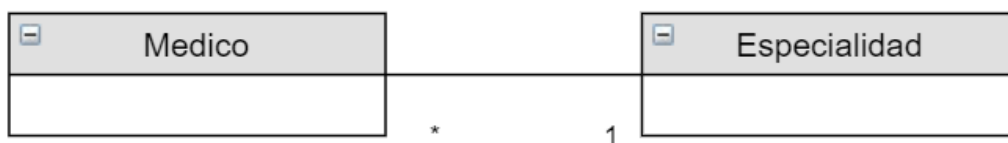


Imagen 1. Entidades médico y especialidad.
Fuente: Desafío Latam

2. El médico tiene un nombre, rut y dirección.

- **Paso 2:** Definimos entonces los atributos de médico en la tabla.

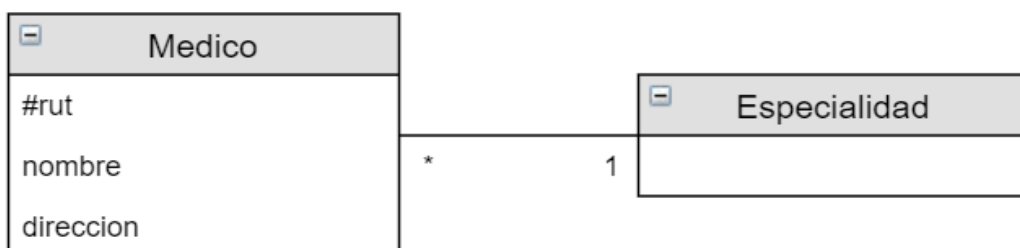


Imagen 2. Agregando atributos a Médico.
Fuente: Desafío Latam

3. La especialidad tiene un código y una descripción.

- **Paso 3:** Definimos los atributos de la tabla Especialidad.

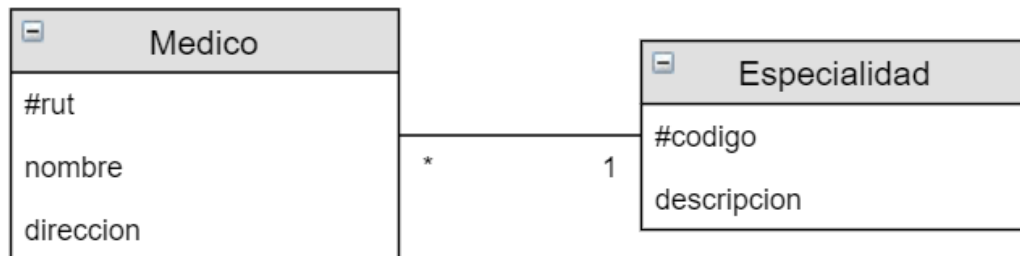


Imagen 3. Atributos de Especialidad.
Fuente: Desafío Latam

4. El médico realiza consultas a distintos pacientes.

- **Paso 4:** Agregar al diagrama las entidades de consulta y paciente. Para estas entidades debemos analizar que técnicamente existe una relación N:N entre médicos y pacientes. Para generar esta cardinalidad se implementa una tabla intermedia llamada Consulta que hará dicha conexión.



Imagen 4. Entidades de Consulta y Paciente.
Fuente: Desafío Latam



Actividad guiada: Integrando normalización en un modelo de datos

A continuación, trabajaremos en un modelo de datos que nos entregan en el cual tendremos estudiantes, matrículas, cursos y carreras y algún que otro dato de utilidad. Este modelo de datos, se encuentra desnormalizado y está causando algunas sobrecargas innecesarias al momento de ser consultado para extraer información.

El modelo de datos es el siguiente:

Tabla Estudiantes

Matrícula	Nombre	Email	Teléfono	Curso	Carrera
1	Francisco	francisco@francisco.com	56955555555	Introducción al desarrollo web	Emprendedor Digital
1	Francisco	francisco@francisco.com	56955555555	Programación con Ruby	Emprendedor Digital
1	Francisco	francisco@francisco.com	56955555555	Bases de datos	Emprendedor Digital
1	Francisco	francisco@francisco.com	56955555555	Introducción a Ruby on Rails	Emprendedor Digital
2	Felipe	felipe@felipe.com	56944444444	Introducción al desarrollo web	Full Stack JavaScript

La clave primaria (PK) para esta entidad de estudiante será la matrícula.

Análisis de normalización

1. Tenemos los datos principales de un estudiante, el nombre, el email y el teléfono.
2. Además, tenemos el curso y carrera que está realizando.
3. Al estudiante podríamos identificarlo de dos maneras, por un lado el email o la matrícula.

Este modelo no está normalizado, tenemos redundancia o repetición de datos como en el caso del estudiante Francisco.

Aplicemos la 1FN a nuestro modelo de datos

- **Paso 1:** Identificar aquellos campos que pueden ser convertidos a una tabla. En este caso podríamos seleccionar el campo Curso. Un curso tranquilamente puede tener una entidad aparte en la cual esté un identificador único y el nombre en sí de dicho curso.

Entidad Cursos

Curso_id	Nombre_curso	Matrícula
1	Introducción al desarrollo web	1
2	Programación con Ruby	1
3	Bases de datos	1
4	Introducción a Ruby on Rails	1
1	Introducción al desarrollo web	2

Al separar en una nueva entidad los cursos estamos conservando el campo Matrícula. Esto se realiza básicamente para mantener la integridad referencial, dado que cada estudiante debe tener una relación con la entidad de Cursos. Entonces, la matrícula nos servirá para relacionar ambas entidades.

La (PK) para esta entidad pasa a ser Curso_id y Matrícula pasa a ser (FK).

- Paso 2: Modificamos entonces la tabla para estudiantes.

Entidad Estudiantes

Matrícula	Nombre	Email	Teléfono	Carrera
1	Francisco	francisco@francisco.com	56955555555	Emprendedor Digital
2	Felipe	felipe@felipe.com	56944444444	Full Stack JavaScript

Dependencia Funcional

Cuando hablamos de dependencias funcionales nos referimos a aquellos campos que dependen de otro para existir o funcionar correctamente.

A–B–C:

- ❖ Supongamos que A es una PK.
- ❖ B y C dependen funcionalmente de A para poder funcionar.

Asociando esto a integridad referencial, supongamos que A es Matrícula, B es Nombre y C es Email. Para este caso, el acceso a la información de manera íntegra y correctamente funcional depende de la existencia de A.

Dependencia Transitiva

La dependencia transitiva se refiere a que los campos de una base de datos no dependen exactamente de otro para poder funcionar.

A > B–C:

- ❖ A es la primary Key.
- ❖ B depende funcionalmente de A.
- ❖ C no depende funcionalmente de A pero sí de B.
- ❖ Como B depende de A automáticamente C también obtendrá esta dependencia aunque no sea directa.

En la entidad de Cursos el campo el curso en sí depende funcionalmente de su identificador único, en este caso Curso_id. Sin embargo, el curso no depende funcionalmente de Matrícula.

Apliquemos la 2FN a nuestro modelo de datos

Para aplicar la segunda forma normal en un modelo de datos debemos eliminar las dependencias transitivas como es el caso de la entidad de cursos. Dicha dependencia está en el campo Matrícula que se comporta como una clave foránea.

Para lograr la 2FN agregamos en una nueva tabla esta relación.

Tabla intermedia

Curso_id	Matrícula
1	1
2	1
3	1
4	1
1	2

Esta tabla intermedia surge del análisis de cardinalidad dado que, un estudiante puede tener muchos cursos y un curso puede tener muchos estudiantes. Entonces, esta tabla intermedia es la que gestiona a través de claves foráneas dicha relación.

Apliquemos la 3FN a nuestro modelo de datos

Recordemos que debemos satisfacer la 2FN de manera inicial.

En la entidad de Estudiantes existe otra dependencia transitiva en el campo de Carrera. Esto dado que la matrícula es la clave primaria del estudiante pero no de la carrera en sí, ante ese análisis podríamos entonces convertir el campo de Carrera en una nueva entidad con su respectiva clave primaria.

Entidad Carreras

Carrera_id	Carrera
1	Emprended or Digital
2	Full Stack JavaScript

Aplicando esto, ahora debemos relacionar las carreras con la entidad Estudiantes.

Entidad modificada de Estudiantes

Matrícula	Nombre	Email	Teléfono	Carrera_id
1	Francisco	francisco@francisco.com	56955555555	1
2	Felipe	felipe@felipe.com	56944444444	2

Reflexión

Si la entidad de estudiantes no la modificamos y dejamos la carrera como un campo más no como clave foránea nos podríamos enfrentar al caso donde existan 1000 registros y el nombre de la carrera se repita N cantidad de veces. El objetivo de la normalización establece que debemos evitar la redundancia de los datos.

A continuación te compartimos un diagrama representativo de estas tablas.

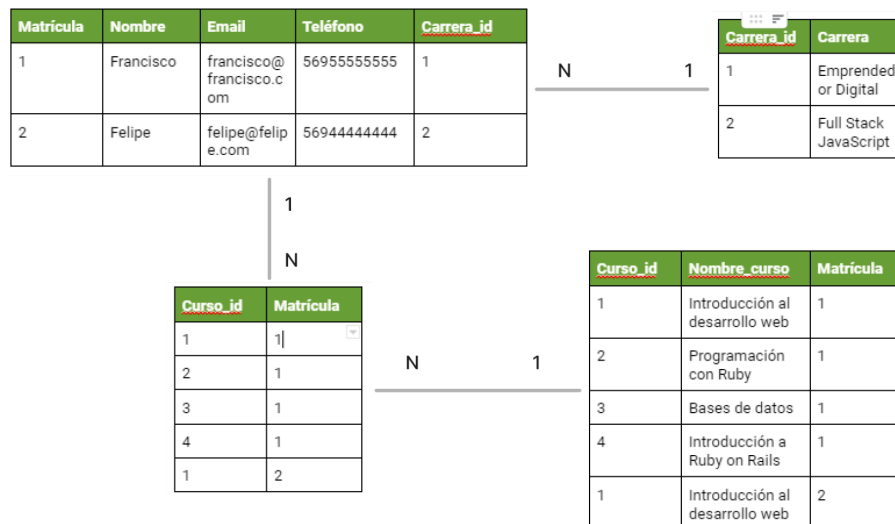


Imagen 5. Diagrama entidad relación
Fuente: Desafío Latam



¡Manos a la obra! - Agregando cardinalidad entre Consulta y Paciente

En el diagrama anterior del ejercicio “**Aplicando modelado y cardinalidad en modelos de datos**” deberás incorporar la nomenclatura al modelo lógico para que la relación Consulta y Paciente sea entendible para otros desarrolladores y el negocio.



Nota: El ejercicio guiado y el propuesto forman parte de los requerimientos iniciales para el Desafío que se realizará en la siguiente sesión.

Solución ejercicio propuesto

