

HIBERNATE: RELACIONES ENTRE TABLAS

Por Mikel Jorge y Eneko Martínez

Vamos a completar la base de datos de la actividad UT3_A1 CRUD. Añadiremos las relaciones que dejamos fuera con sus nuevos atributos.

Antes de empezar...

Necesitamos haber completado la actividad anterior UT3_A1 CRUD. Puedes usar la solución disponible en Moodle. Asegúrate de configurar adecuadamente los parámetros de conexión a la base de datos.

Creación del proyecto

Partiremos del proyecto de la actividad anterior, completándolo con las partes que dejamos fuera. Usa el proyecto anterior renombrándolo o copia el código en un proyecto nuevo, como prefieras.

¿Qué queremos que haga nuestro programa?

Nuestra aplicación va a crear en una base de datos vacía las tablas que representen las entidades y relaciones de requerida en los esquemas proporcionados previamente, con la estructura y tipos de datos del cliente. En esta actividad **tendremos en cuenta las relaciones existentes entre tablas**. Por lo tanto, **incluiremos** aquellos atributos que se creen para cumplir las relaciones así como las restricciones que se puedan derivar de ello.

Posteriormente, en el programa principal daremos opción, mediante un menú de texto, a probar las operaciones CRUD para cada una de las relaciones y tablas que hayamos creado.

1. Información previa relativa a la base de datos del cliente

Nos proporcionan los esquemas de la base de datos que requiere el cliente.

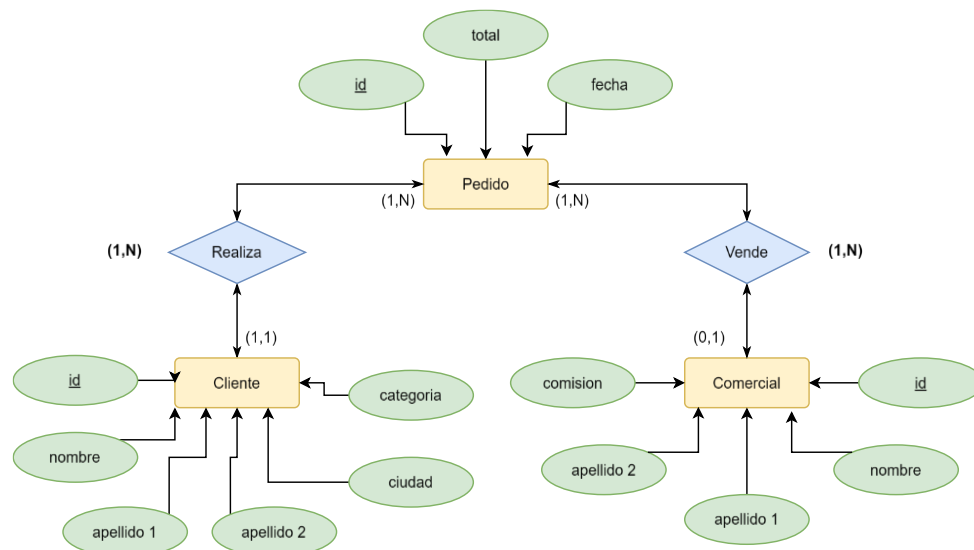


DIAGRAMA 1: CONCEPTUAL (E/R)

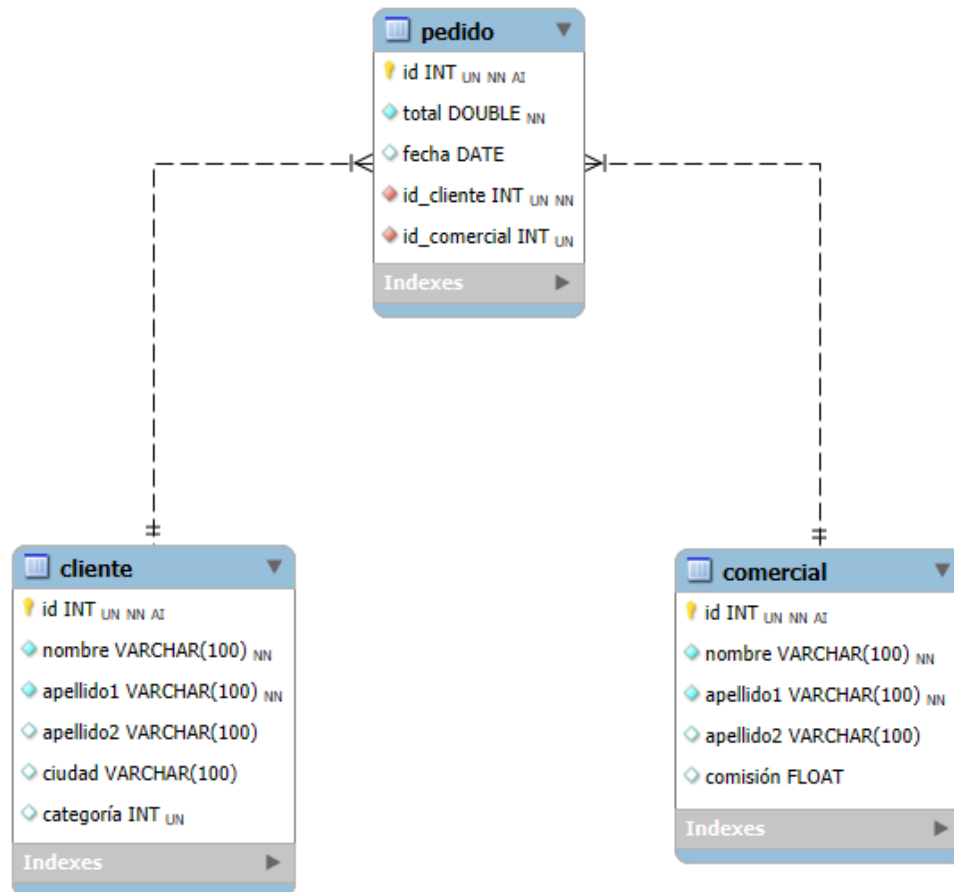


DIAGRAMA 2: RELACIONAL

Nota: 'UN' especifica un entero sin signo (UNSIGNED).

2. Clase de conexión y archivo de configuración de Hibernate

No varían con respecto a la actividad anterior. Eso sí, en caso de usar como base la solución a la actividad anterior entregada por el profesor, tendremos que revisar y en caso de necesidad cambiar los parámetros adecuados dentro del archivo de configuración de Hibernate que está en la carpeta *resources*.

3. Clases modelo

Tendremos que añadir a las clases del modelo las relaciones y claves foráneas necesarias para cumplir con lo pedido en ellos esquemas de la base de datos, de manera similar a lo visto en los apuntes. **Tendremos que usar obligatoriamente fetch_type Lazy y cascade All.**

Revisa bien en los apuntes cómo debemos gestionar cada tipo de relación y luego refléjalo en la actividad. Por último, tendremos que tener en cuenta y decidir cómo afecta a nuestro modelo y a las tablas correspondiente en la base de datos las cardinalidades de las relaciones.

4. Clases DAO

Añade los métodos necesarios para que podamos acceder a todos los datos de las clases modelo a través de las relaciones entre tablas.

5. Clases de servicios

Añade los métodos necesarios para que podamos acceder a todos los datos de las clases modelo a través de las relaciones entre tablas. Piensa si y cuándo los accesos a la base de datos van a necesitar el uso de transacciones.

6. Programa principal

Modifica el programa principal para que las pruebas incluyan y muestren los datos contenidos en los nuevos atributos de clase generados por las relaciones. Debes ir indicando en pantalla paso a paso lo que estás haciendo y probando en cada momento.