|  |
| --- |
| PROYECTO FINAL  SQL |
|  |
| 2022  SERGIO EMANUEL PRELLEZO  Coderhouse |

**Índice**

[Introducción 3](#_Toc105609236)

[Objetivo 3](#_Toc105609237)

[Situación problemática 3](#_Toc105609238)

[Modelo de negocio 4](#_Toc105609239)

[Diagrama de entidad relación 5](#_Toc105609240)

[Tablas 7](#_Toc105609241)

[Listado 7](#_Toc105609242)

[Descripción 8](#_Toc105609243)

[Scripts de creación de los objetos de la base de datos 12](#_Toc105609244)

[Esquemas relacionales - *Reverse Engineer* 12](#_Toc105609245)

# Introducción

Este informe constituye el proyecto final para el curso de SQL dictado por Coderhouse.

Recientemente acabo de terminar la carrera de “Desarrollo Full Stack” y para la misma realicé un proyecto de ecommerce, por lo que decidí tomar como referencia ese trabajo para la elección de mi situación problemática y modelo de negocio.

# Objetivo

Crear una base de datos, en la cual se implementará el modelo relacional para representar procesos basados en un modelo de negocio propio, con dataset ficticio y derivado de un proyecto anterior de ecommerce.

Se desarrollarán objetos e implementarán los procesos técnicos que requiere el mantenimiento de una base de datos.

También se implementarán consultas SQL que permitan la generación de informes de diferente índole y alcance.

# Situación problemática

Nos encontramos ante un comercio electrónico, más específicamente, una tienda de bicicletas y artículos de ciclismo llamada **Mammoth**.

Se plantea la necesidad de dar soporte a la app de la empresa generando la persistencia de la información en una base de datos SQL. En mi caso utilizaré a MySQL como sistema gestor de base de datos (**SGBD**).

Las principales necesidades del comercio son:

* guardar información de contacto de sus clientes
* guardar preferencias de los clientes tales como favoritos y suscripciones con el fin de poder conocer mejor a los usuarios y generar análisis que den origen a potenciales oportunidades de venta.
* almacenar los carritos de compra activos
* ofrecer su catálogo de productos
* capacidad de administrar los productos y controlar su stock
* guardar y llevar un control de los pedidos y sus respectivos estados
* guardar los detalles de facturación y posibilitar diversos análisis sobre la misma.

# Modelo de negocio

Como comenté anteriormente, estamos frente a una tienda online (**Mammoth**).

La plataforma permite visualizar el catálogo de productos junto con la posibilidad de ver en detalle cada uno de ellos, junto a su actual precio de venta y su disponibilidad de stock.

Los productos pertenecen a determinadas categorías del comercio (cada producto se asocia con una única categoría), tales como bicicletas, accesorios, indumentaria, etc. También se informa el país de procedencia de los mismos y quién es el proveedor en cada caso (cada producto tiene asociado un único proveedor), aunque esta última información es solo visible por los miembros de la empresa.

Para proceder a interactuar con la plataforma en cuanto a la selección de productos, selección de favoritos y suscripciones, es necesario que el usuario complete un registro brindando información de contacto y facturación. En un principio el comercio va a trabajar con clientes locales, por lo que la información precargada en las tablas de ciudades y provincias corresponden a la República Argentina.

El usuario puede elegir distintos productos y cantidades de los mismos para su carrito de compras, el cuál va a ser almacenado por la empresa durante un periodo de tiempo que se considere razonable. Carritos más antiguos que dicho período procederán a borrarse por cuestiones de optimización de almacenamiento del servidor.

Si se elige más de una unidad de un determinado producto, se agrupa dicha cantidad, dando origen a un detalle de carrito o pedido optimizado sin repeticiones de un mismo artículo.

Una vez que el usuario confirma un pedido, se procede a cargar su detalle en base al detalle del carrito que da origen a la orden de venta y posteriormente si todo ocurre correctamente, se procede a eliminar dicho carrito ya que carece de sentido mantenerlo persistente. De esta manera cada usuario puede tener un único carrito activo a la vez.

También previo a generar la orden de venta, se chequea la disponibilidad de stock y de ser correcta, el producto actualiza su stock.

Al realizar la compra, el usuario puede escoger entre distintos tipos de envío y también diferentes opciones de pago. Existe una relación biunívoca entre la orden de venta y la factura emitida, por lo que el detalle de la factura es exactamente el mismo que la orden de venta relacionada y por ende se toma de ella sin redundar información.

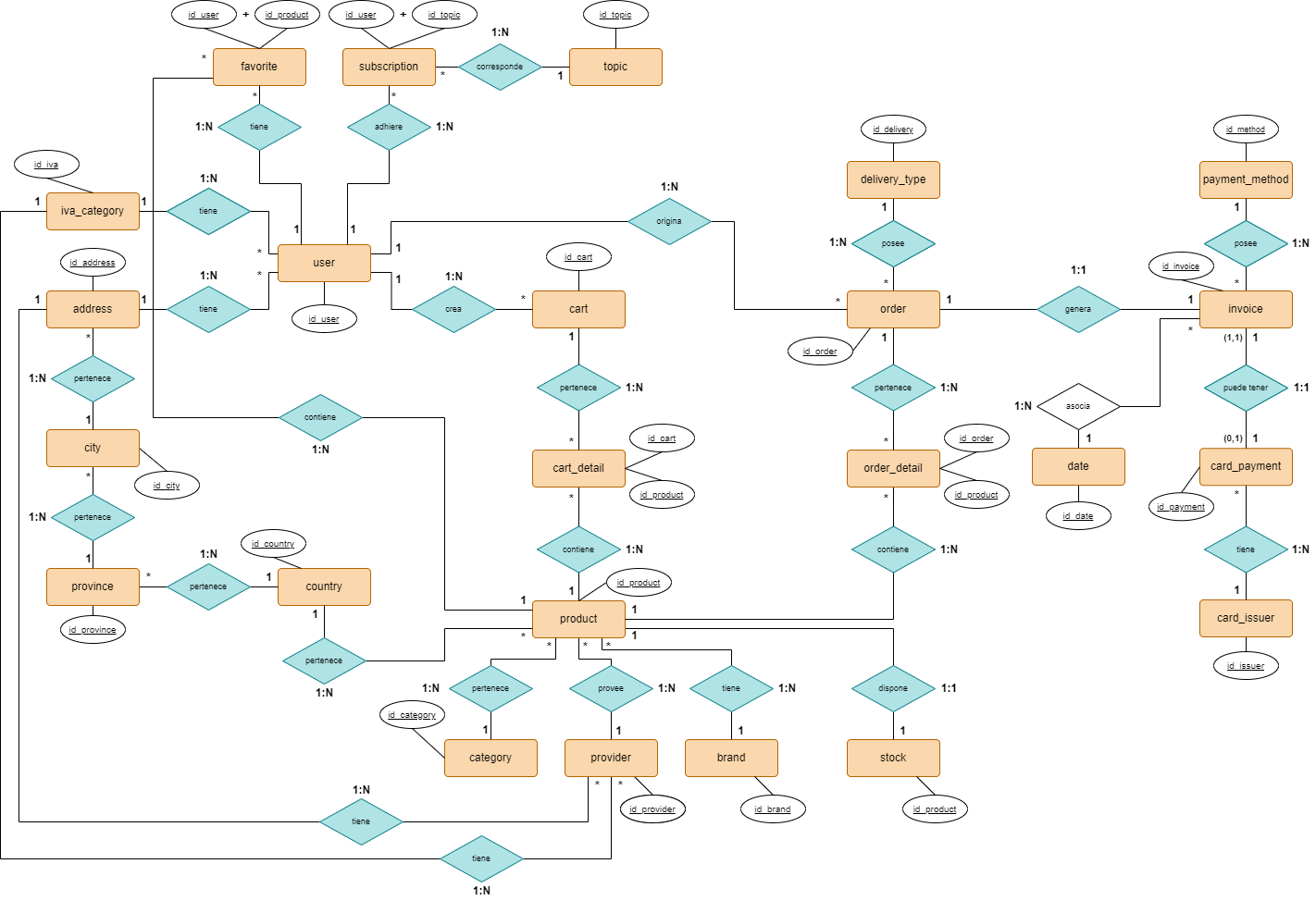
Finalmente, un usuario registrado puede tener múltiples productos favoritos y también se puede suscribir a múltiples temáticas de interés.

# Diagrama de entidad relación

En la siguiente hoja se muestra el diagrama de entidad relación (**DER**) correspondiente al modelo de base de datos desarrollado **mammoth**.

El diagrama posee la siguiente terminología estándar:

* Rectángulos: representan conjuntos de entidades.
* Elipses: representan atributos. En general se colocan solamente los nombres subrayados de los campos que son llaves primarias (**PK**).
* Rombos: representan acciones que permiten relaciones entre tablas.
* Líneas: unen atributos a conjuntos de entidades y conjuntos de entidades a conjuntos de relaciones. Representan la unión entre acciones y entidades
* Cardinalidad: Representada en cada extremo de la relación y también sobre cada acción.



# Tablas

A continuación, se encuentra el listado de tablas incluido en la base de datos **mammoth** y luego la descripción de cada una de ellas

## Listado

1. *user*
2. *subscription*
3. *topic*
4. *favorite*
5. *iva\_category*
6. *address*
7. *city*
8. *province*
9. *country*
10. *cart*
11. *cart\_detail*
12. *product*
13. *category*
14. *brand*
15. *provider*
16. *stock*
17. *order*
18. *order\_detail*
19. *delivery\_type*
20. *invoice*
21. *payment\_method*
22. *card\_payment*
23. *card\_issuer*
24. *date*

## Descripción

















































# Scripts de creación de los objetos de la base de datos

Se adjunta a este documento los scripts de creación correspondientes en archivos de extensión sql (mammoth-CREATE.sql).

# Esquemas relacionales - *Reverse Engineer*

Se adjunta a este documento el archivo *diagram-mammoth.mwb* con el diagrama EER obtenido aplicando ingeniería inversa en el cliente SGDB MySQL Workbench.

A continuación se puede observar una visualización general.

