Testing For Learning S.A   
Informe sobre la integridad de los datos almacenados en la base de datos del Ecommerce Emercado, y las secuencias propuestas para los casos de prueba 001 y 002.

Nombre y apellido: Nahuel Arsen Balian Guerrero

Fecha: 24/11/22

Informe de base de datos

EMercado.

### 

### Índice

[Índice 0](#_Toc120389798)

[| Historial de Cambios 1](#_Toc120389799)

[| Introducción y alcance 2](#_Toc120389800)

[| Reporte de integridad de datos. 3](#_Toc120389801)

[| Casos de prueba y sentencias propuestas. 6](#_Toc120389802)

### 

### 

### | Historial de Cambios

*[En general todos los documentos se versionan y se incluye dentro del mismo una reseña de la historia de cambios realizada en donde se suele especificar la versión del documento, la fecha de su realización, quién es responsable de elaborar ese documento y algún comentario sobre las modificaciones realizadas al mismo. De esta manera, alguien que lee un documento recientemente actualizado no tiene necesidad de leer todo de nuevo sino que puede ir directamente a ver la sección o secciones que sufrieron cambios. Se adjunta más abajo una tabla en donde se podría colocar esta información para facilitar su visualización.]*

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Responsable | Resumen del cambio |
| 1.0 | 24/11/2022 | Nahuel Balian | Se agregó la caratula, introducción y alcance, reporte de integridad de datos, y casos de prueba. |
|  |  |  |  |
|  |  |  |  |

### | Introducción y alcance

*[*Este informe tratara sobre el los resultados del estudio de verificación de integridad de datos realizado sobre la base de datos de la pagina ecommerce “Emercado”, para la empresa Jóvenes a Programar, con la finalidad de verificar la correctitud y completitud de la base de datos.  
  
Este producto se trata de un sitio web escrito utilizando HTML, CSS, y Javascript, en el cual se comercian una variedad de items, desde autos, juguetes, e incluso mobiliario.  
  
Este informe, en particular, girará en torno a la verificación de que la base de datos tenga integridad de dominio, de unicidad de clave, y de integridad referencial. A su vez, este documento también abarcará las secuencias SQL propuestas para realizar los casos de prueba ID 0001 y 0002.

### | Reporte de integridad de datos.

Se encontraron una variedad de incidencias relacionadas a la integridad de los datos almacenados en la base de datos. A continuación, se detallaran las incidencias en la integridad de los datos, y se detallaran las incidencias encontradas en los datos almacenados.

***DATOS PRODUCTO***

INCIDENCIAS INTEGRIDAD:  
Admite nulls en costo, moneda, descripción, y cantidad de stock. Esto es incorrecto, ya que la pagina ha de contener info en cantidad, descripción, moneda, y costo.

Existen conjuntos de datos cuyo valor en precio es nulo (código 28, 29, y 30), lo que es incorrecto, puesto que la pagina ha de contar con esta info, pero posible, debido al dominio.  
Pasa igual con la columna Cantidad\_stock en los códigos 20, 21, y 22.

INCIDENCIAS DOMINIO:  
La columna cantidad stock tiene dominio numeric(5,2), lo que significa que admite números de 5 cifras con 2 unidades después de la coma. Esto es incorrecto, ya que no se puede tener menos de una unidad de algo en este contexto.

Esto causa errores en los datos almacenados de esta columna, pues hay muchos conjuntos de datos con números con coma en la tabla.

Según los requisitos del cliente, la descripción del producto debería tener un máximo de 50 caracteres. La tabla admite 500, por lo que el dominio es erróneo. Aun asi, es recomendable reconsiderar este requisito ya que 50 caracteres es muy poco para dar una descripción adecuada en una pagina de ecommerce.  
  
  
***DATOS PUNTUACIÓN***

No se encontraron incidencias en la integridad de datos o en los datos almacenados.

***DATOS USUARIO***

INCIDENCIAS DOMINIO:

Admite nulls en apellido y email.

En esta tabla parece tener datos válidos, a pesar de los errores de dominio que tiene el diseño.

***DATOS CATEGORIA\_PRODUCTO***

No se encontraron incidencias en la integridad de datos o en los datos almacenados.

***DATOS CATEGORIA***

INCIDENCIAS DOMINIO:

Permite nulls en descripción. Aunque la propuesta y ESRE no definen que las categorías deban de tener una descripción, por motivo de consistencia remover el null, ya que, si todas las categorías tienen descripción, agregar una nueva categoría sin describirla seria inconsistente.

***DATOS IMAGEN\_CATEGORIA***  
  
INCIDENCIAS DOMINIO:  
Admite nulls en nombre.  
Sin errores aparentes en los datos almacenados. Todos los valores de la columna “Datos” son iguales.

***DATOS IMAGEN PRODUCTO***

INCIDENCIAS DOMINIO:

Admite nulls en nombre.  
Sin errores aparentes en los datos almacenados. Todos los valores de la columna “Datos” son iguales.

***DATOS PRODUCTO\_CARRITO***   
  
INCIDENCIAS INTEGRIDAD DE REFERENCIAL:

La columna “Cod\_Carrito” debería ser una FK hacia la columna “Codigo” de la tabla carrito, pero no lo es.

INCIDENCIAS DOMINIO:

La columna “Cod\_Carrito” admite nulls.  
Uno de los datos de los elementos de esta tabla pose un valor null en la columna FK “Cod\_Carrito”. Esto es incorrecto, ya que no solo este columna no debería admitir nulls, esta es una FK que referencia una PK de otra tabla, por lo que no debería ser posible que sea null. Esto se debe a que esta FK no esta finalizada, y no referencia la llave primaria a la que debería referenciar.  
  
El conjunto de datos cuyo Cod\_Carrito es 4, posee una cantidad de “0” en su carrito. Esto es erróneo, ya que el carrito no debería admitir la posibilidad de comprar 0 unidades de un producto.

***DATOS PRODUCTO\_RELACIONADO***

INCIDENCIAS DE INTEGRIDAD REFERENCIAL:

Los datos almacenados en esta tabla referencian columnas inexistentes de la tabla Producto. Son FK funcionales, puesto que el diagrama SQL muestra una referencia hacia y desde la tabla producto, pero no parece haber una referencia en si.

***DATOS CARRITO***

INCIDENCIAS DOMINIO:

Admite nulls en código usuario, código envio, código pago, fecha, calle, numero, esquina, y país.  
Los datos en cod\_Usuario del código 2, son nulos  
  
INCIDENCIAS DE INTEGRIDAD REFERENCIAL:

La columna “Cod\_Usuario” debería ser una FK que referencie la columna “Codigo” de la tabla usuario. No lo es.

***DATOS PAGO***

La mayoría de los conjuntos de datos ingresados poseen valores nulls. Esto es correcto, ya que se ingresan valores según el método elegido, por lo que es permitido que uno u otro quede null.  
El dominio para nro tarjeta de crédito y cuenta bancaria debería ser INT, no nvarchar, puesto que nvcarchar permite el ingreso de valores alfanuméricos.

***DATOS METODO\_ENVIO***

No se encontraron incidencias en la integridad de datos o en los datos almacenados.

### | Casos de prueba y sentencias propuestas.

Para el caso de prueba 0002, se resolvió establecer una sentencia UPDATE que actualizara los datos de “Cantidad\_Stock” del producto código 47, restándole 4, para así tener la cantidad restante de stock adecuada después de la compra de 4 articulos.

UPDATE producto SET cantidad\_stock = cantidad\_stock -4 WHERE codigo = 47;

Para el caso de prueba 0002, se resolvió estructurar una sentencia que mostrará todas las categorías de los productos de la pagina (A travez de los datos de la columna “Nombre” de la tabla “Categoria”), y contará la cantidad de artículos que contiene cada categoría.

Para esto se utilizo una sentencia Count que contara los códigos de categoría como “Productos” de la tabla “Categoria” uniendo a los datos de la columna “Categoria\_producto” con los datos de “Cod\_Categoria” y ordenandolos por el nombre.

SELECT nombre, COUNT (COD\_CATEGORIA) as productos FROM CATEGORIA left join categoria\_producto on codigo = cod\_categoria group by categoria.nombre