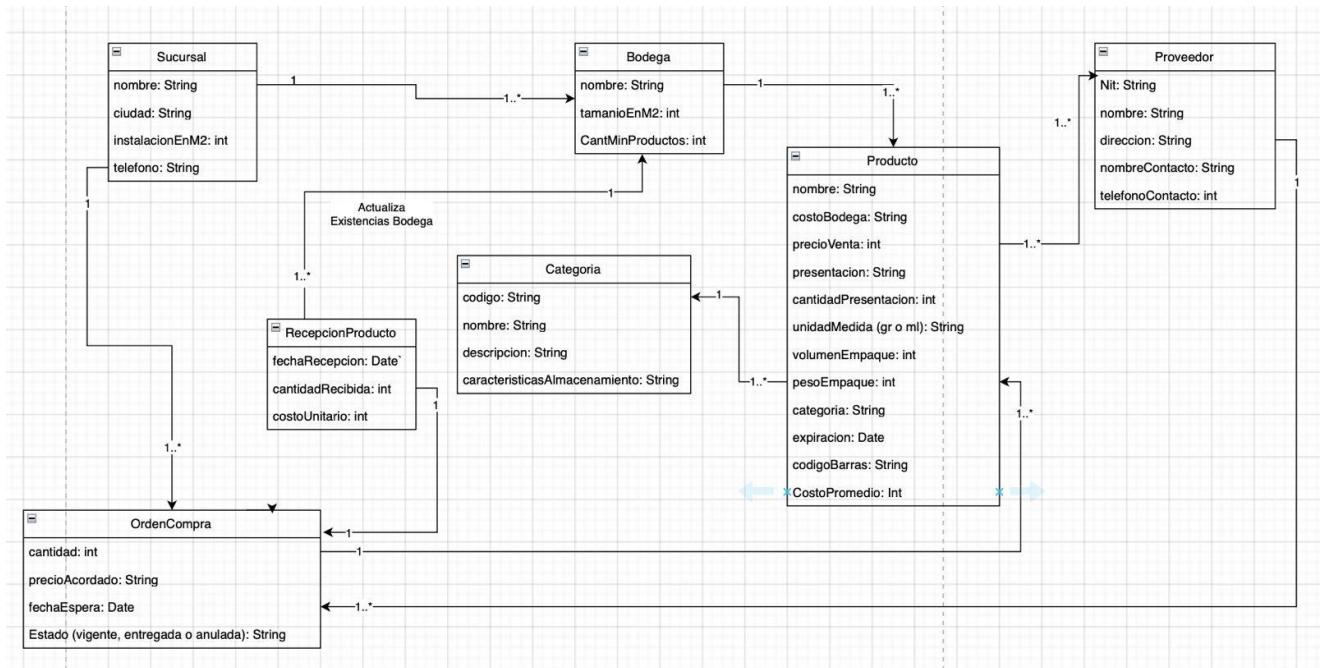
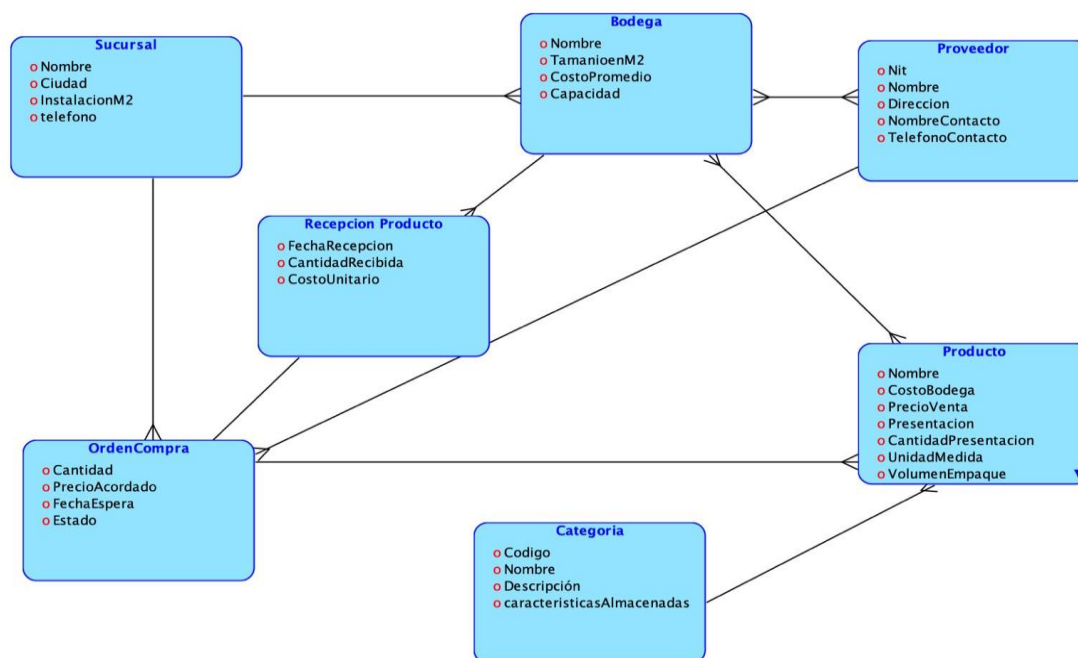


Proyecto 1 – Sistrans

UML:



Modelo E/R



1. Identificación de Entidades:

Entidad: Sucursal

- En el modelo UML, una sucursal es una entidad que representa una unidad de operación en diferentes ciudades. En el modelo relacional, los atributos de la entidad son nombre, ciudad, Instalación en m2 y teléfono, para todos se tiene de restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío. Específicamente ID_SUCURSAL que es nuestra primary key (la llave identificadora) esta tiene como restricciones SA que significa que desempeña diferentes funciones para que el funcione y que todos los valores de la columna tienen que tener un valor.

Entidad: Bodega

- En el modelo UML, una bodega es una entidad que representa el lugar donde se almacenan los productos antes de ser enviados a su respectiva sucursal. Tiene atributos como nombre, tamaño en m2, costo promedio, capacidad, y el id de la sucursal a la que está asociada, para todos se tiene como restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío. Específicamente ID_BODEGA que es nuestra primary key (la llave identificadora) esta tiene como restricciones SA que significa que desempeña diferentes funciones para que el funcione y que todos los valores de la columna tienen que tener un valor.

Entidad: Producto

- En el modelo UML, un producto es una entidad que representa todos los artículos disponibles en el supermercado. Tiene atributos como nombre, costo en bodega, precio de venta, presentación, cantidad, unidad de medida, peso empaque, expiración y código de barras y el id de la categoría a la que está asociada, para todos se tiene como restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío. Además en código de barras se tiene de restricción ND la cual nos indica que los valores dentro de esta columna no pueden ser repetidos ya que son únicos al producto que están representando. Específicamente ID_PRODUCTO que es nuestra primary key (la llave identificadora) esta tiene como restricciones SA que significa que desempeña diferentes funciones para que el funcione y que todos los valores de la columna tienen que tener un valor.

Entidad: Categoría

- En el modelo UML, la categoría es la entidad encargada de clasificar los productos según su tipo (perecederos, no perecederos) . Tiene atributos como código, nombre, descripción y características de almacenamiento, para todos se tiene de restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío. Específicamente ID_CATEGORIA que es nuestra primary key (la llave identificadora) esta tiene como restricciones SA que significa que desempeña diferentes funciones para que el funcione y que todos los valores de la columna tienen que tener un valor.

Entidad: Proveedor

- En el modelo UML, un proveedor es una entidad que representa las compañías que suministran los productos a las sucursales. Tiene atributos como NIT, nombre, dirección, nombre del contacto y teléfono de contacto., para todos se tiene de restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío. Además, para el atributo NIT se restringe con ND (not duplicate) ya que cada código de nit es único para cada empresa. Específicamente ID_PROVEEDOR que es nuestra primary key (la llave identificadora) esta tiene como restricciones SA que significa que desempeña diferentes funciones para que el funcione y que todos los valores de la columna tienen que tener un valor.

Entidad: OrdenCompra

- En el modelo UML, una orden de compra es una entidad que registra la adquisición de productos de los proveedores. Tiene atributos como cantidad, precio acordado, fecha de espera, estado, id del proveedor a quien le corresponde la orden de compra e id del producto que se va a adquirir, para todos se tiene de restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío. Además, para el atributo precio acordado se restringe con ND (not duplicate) ya que un precio de una orden de compra no se puede repetir. Específicamente ID_ORDENCOMPRA que es nuestra primary key (la llave identificadora) esta tiene como restricciones SA que significa que desempeña diferentes funciones para que el funcione y que todos los valores de la columna tienen que tener un valor.

Entidad: RecepciónProducto

- Esta entidad registra la recepción de productos en una bodega específica. Tiene atributos como fecha de recepción, cantidad recibida, costo unitario, id de la bodega donde pertenece e id del producto, para todos se tiene como restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío. Específicamente ID_RECEPCIÓNPRODUCTO que es nuestra primary key (la llave identificadora) esta tiene como restricciones SA que significa que desempeña diferentes funciones para que el funcione y que todos los valores de la columna tienen que tener un valor.

Entidad: Orden_Producto

- Esta entidad registra la relación que tienen el producto y la orden de compra. Los atributos que tienen son las llaves foráneas que llaman a sus ids, es decir el id de producto y el id de orden de compra. Para estas dos llaves se tiene como restricción NN (not null) el cual especifica que ningún elemento de esa columna debe estar vacío.

Entidad: Producto_proveedor

2. Relaciones:

Relación: Sucursal-Bodega

- **Descripción:** Una sucursal tiene una o varias bodegas, lo que es una relación 1 a muchos. Ya que la sucursal puede pedir nuevo inventario de varias bodegas si hay un imprevisto con alguna de ellas.
- **Transformación:** Esta relación se maneja a través de la clave foránea id_sucursal en la tabla Bodega, que referencia la tabla Sucursal.

Relación: Bodega-Producto

- **Descripción:** Una bodega puede almacenar varios productos y uno en varias bodegas, una relación N. No tendría sentido que una bodega pueda almacenar solo uno de ellos puede almacenar diferentes si tiene la capacidad necesaria para hacerlo.
- **Transformación:** Esta relación podría requerir una tabla intermedia para manejar la relación muchos a muchos. En este caso se maneja mediante la inclusión de id_bodega y id_producto en la tabla RecepciónProducto.

Relación: Producto-Categoría

- **Descripción:** Un producto pertenece a una categoría y una categoría puede tener múltiples productos. Un producto obligatoriamente debe pertenecer a una categoría y múltiples productos pueden pertenecer a múltiples categorías como por ejemplo “refrigerados” que son productos que pueden ser almacenados juntos en una bodega refrigerada
- **Transformación:** Se maneja a través de la clave foránea id_categoria en la tabla Producto, que referencia la tabla Categoría.

Relación: Proveedor-Producto

- **Descripción:** Un proveedor puede suministrar uno o varios productos y uno o varios proveedores pueden suministrarlo (ambos 1..*). Es lógico que varios proveedores puedan adquirir un producto si se tienen inconvenientes con alguno de estos. Por otra parte, un proveedor puede proveer múltiples productos pues no hay una razón para restringir al proveedor a suministrar únicamente un producto
- **Transformación:** Esta relación se maneja en la tabla OrdenCompra donde id_proveedor y id_producto se relacionan con las tablas correspondientes.

Relación: Orden de Compra-Recepción de Producto

- **Descripción:** Es verdad que una orden de compra puede incluir múltiples productos, pero una recepción de producto se asocia a una única orden de compra. Al recibir un grupo de productos estos van a estos asociados a solo una orden de compra. Al recibir los productos no se pueden combinar productos de otras órdenes de compra
- **Transformación:** Esta relación se maneja mediante las claves foráneas en las tablas correspondientes (id_ordenCompra en RecepciónProducto).

Relación: Sucursal-Orden de compra

- **Descripción:** una Orden de compra debe estar dirigida a 1 sucursal. Por otro lado, una sucursal puede tener varias órdenes de compra. Se pueden haber realizado varias órdenes de compra por diferentes sucursales, una sucursal puede tener 1 o muchas órdenes de compra pues estas pueden estar asociadas a diferentes proveedores.
- **Transformación:** Esta relación se maneja mediante las llaves foráneas en la tabla orden Compra referenciando el id sucursal).

Relación: Bodega-Recepción Producto

- **Descripción:** una bodega puede tener varias recepciones de productos y una recepción de productos está asociada a una bodega. Las recepciones de productos se hacen por bodega. Una bodega puede múltiples pedidos de otra manera cuando se acaben no se podría reabastecer
- **Transformación:** Esta relación se maneja mediante las llaves foráneas en la tabla recepcionProducto referenciando el id bodega).

Relación: Orden de Compra-Producto

- **Descripción:** Una orden de compra puede tener múltiples productos, sin embargo, y un producto puede aparecer en diferentes órdenes de compra siempre y cuando sean diferentes unidades (si es un televisor que ese mismo televisor no aparezca en otra orden de compra, sino que sea otra unidad del mismo modelo de televisor). Un almacén puede pedir múltiples productos en una misma orden de compra. Y un producto puede ser pedido por múltiples órdenes de compra.
- **Transformación:** Esta relación se maneja mediante una tercera tabla que tiene como atributos una llave foránea que referencie el id del producto y una llave foránea que represente el id de la orden de compra.

Relación: Orden de Compra-Proveedor

- **Descripción:** Una orden de compra solo está asociada a un proveedor mientras que un proveedor puede tener múltiples órdenes de compra. Si una orden de compra estuviera asociado a múltiples proveedores tocaría especificar que productos vienen de que proveedor y a que orden de compra pertenecen complicando las tablas. Y un proveedor no tiene razón para estar restringido a tener un solo orden de compra.
- **Transformación:** Esta relación se maneja mediante las llaves foráneas en la tabla OrdenCompra referenciando el id_proveedor).

3. Proceso de Normalización:

- **Primera Forma Normal (1FN):** Cada tabla contiene solo valores atómicos; es decir, no hay conjuntos o listas de valores en una sola celda.

- **Segunda Forma Normal (2FN):** Todas las tablas están en 1FN, y todos los atributos que no son clave dependen completamente de la clave primaria.
- **Tercera Forma Normal (3FN):** Todas las tablas están en 2FN, y no hay dependencia transitiva de los atributos que no son clave primaria respecto a la clave primaria.

Para la tabla sucursal, todos los atributos contienen valores atómicos: nombre, ciudad, instalacionEnM2, telefono, es decir que cumple con 1NF. Ahora bien, todos los atributos dependen completamente de la llave principal: id_sucursal por lo que cumple con 2NF. Asimismo no existen dependencias transitivas y por tanto cumple con 3NF. Por último, la llave primaria es la única superllave (cumple con BCNF).

Para la tabla bodega, todos los atributos contienen valores atómicos: nombre, tamanoEnM2, costoPromedio, capacidad, lo que indica que cumple con 1NF. Asimismo, todos estos atributos dependen netamente de la llave principal: id_bodega y cumple con 2NF. No existen dependencias transitivas (3NF), y la llave primaria es la única superllave (BCNF).

Para la tabla producto, los atributos contienen valores atómicos: nombre, costoBodega, precioVenta, presentación, cantidadPresentacion, unidadMedida, volumenEmpaque, pesoEmpaque, expiracion, codigoBarras (1NF). De igual manera, todos los atributos dependen directamente de la llave principal: id_producto (2NF). También no se reflejan dependencias transitivas (3NF) y además la llave primaria es superllave (BCNF).

Para la tabla categoría, todos los atributos contienen valores atómicos: código, nombre, descripcion, caracteristicasAlmacenamiento, es decir que cumple con 1NF. Igualmente, todos los atributos dependen de la llave principal: id_categoria y por tanto respeta la segunda forma normal. No existen dependencias transitivas. Y la llave primaria es la única superllave (BCNF).

Para la tabla proveedor, todos los atributos contienen valores atómicos: nit, nombre, direccion, nombreContacto, telefonoContacto, es decir que cumple con 1NF. Ahora bien, todos los atributos dependen completamente de la llave principal: id_proveedor por lo que cumple con 2NF. Asimismo no existen dependencias transitivas y por tanto cumple con 3NF. Por último, la llave primaria es la única superllave (cumple con BCNF).

Para la tabla ordenCompra, todos los atributos contienen valores atómicos: cantidad, precioAcordado, fechaEspera, estado, es decir que cumple con 1NF. Ahora bien, todos los atributos dependen completamente de la llave principal: id_compra por lo que cumple con 2NF. Asimismo no existen dependencias transitivas y por tanto cumple con 3NF. Por último, la llave primaria es la única superllave (cumple con BCNF).

Para la tabla RecepcionProducto, todos los atributos contienen valores atómicos: fechaRecepcion, cantidadRecibida, costoUnitario, es decir que cumple con 1NF. Ahora bien, todos los atributos dependen completamente de la llave principal: id_recepcion por lo que cumple con 2NF. Asimismo no existen dependencias transitivas y por tanto cumple con 3NF. Por último, la llave primaria es la única superllave (cumple con BCNF).

Para la tabla orden_Producto, todos los atributos contienen valores atómicos: id_Producto, id_ordenCompra, es decir que cumple con 1NF. Asimismo, todos estos atributos dependen netamente de la llave principal compuesta por: id_producto e id_ordenCompra y cumple con 2NF. No existen dependencias transitivas (3NF), y la llave primaria compuesta es la única superllave (BCNF)

Y por último, para la tabla producto_proveedor, todos los atributos contienen valores atómicos: id_Producto, id_proveedor, es decir que cumple con 1NF. Asimismo, todos estos atributos dependen netamente de la llave principal compuesta por: id_producto e id_proveedor y cumple con 2NF. No existen dependencias transitivas (3NF), y la llave primaria compuesta es la única superllave (BCNF)