

Nicolás Ballén-202310273
María Juliana Ballesteros-202313216
Mauricio Martinez-202314461

Proyecto SuperAndes

Entrega 2-Implementación

Grupo C7

Contexto

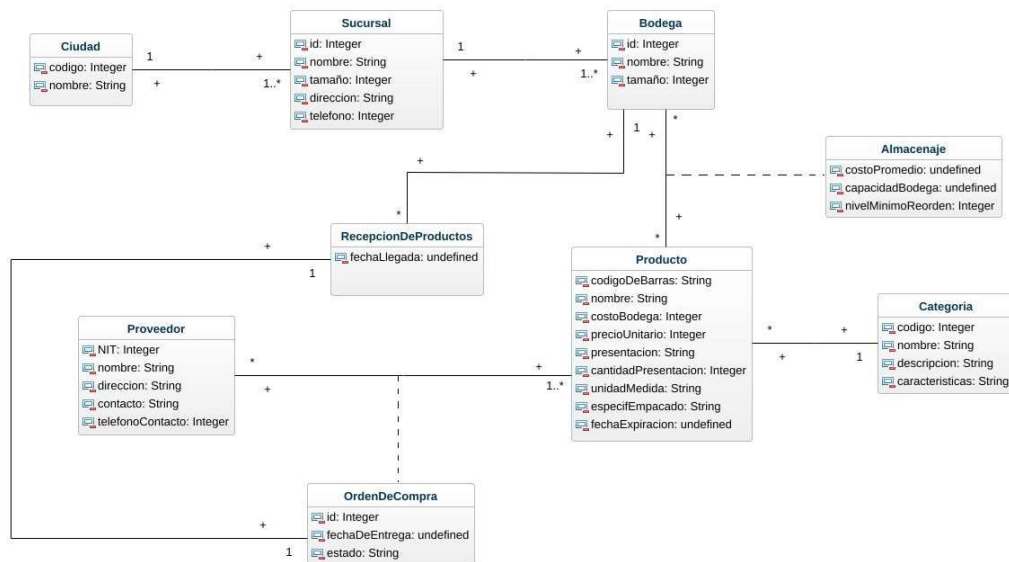
Nos vimos en la tarea de desarrollar una base de datos para la cadena de supermercados SuperAndes. Tras la lectura del informe, nuestro grupo identifico las siguientes entidades que componen el comportamiento y registros del supermercado:

- **Ciudad:** Comprende la ciudad en la que tiene actividad el supermercado.
- **Sucursal:** Comprende las diferentes sucursales del supermercado. Cada sucursal se ubica en una ciudad, habiendo múltiples sucursales por ciudad. Registra su nombre, dirección, teléfono y tamaño.
- **Bodega:** Comprende las bodegas donde se guardan los productos del supermercado. Cada bodega corresponde a una sucursal, pudiendo estas tener múltiples bodegas. Registra su tamaño y nombre.
- **Producto:** Comprende los diferentes productos que se venden en el supermercado. Reconoce su precio de venta, de bodega, código de barras, nombre, presentación, cantidad, unidades de medida, fecha de vencimiento, peso y volumen. Los productos pueden guardarse en múltiples bodegas.
- **Almacenaje:** Comprende los datos de como se amacena X producto en X bodega. Esto reconoce tanto el producto como la bodega en la que se guarda, la cantidad/capacidad de dicho producto en la misma, su costo promedio, su nivel mínimo de reorden (cantidad mínima de unidades que debe tener la bodega antes de solicitar más) y su cantidad.
- **Categoría:** Comprende los tipos de productos que hay en la sucursal. A cada producto le corresponde una categoría, habiendo varios productos por categoría. Esta reconoce una descripción y características de esta.
- **Proveedor:** Comprende a los diferentes proveedores que suministran al supermercado. Cada proveedor puede ofrecer diferentes productos, así como un producto puede ser ofrecido por varios proveedores. Comprenden su NIT, dirección, contacto y teléfono de contacto.
- **Orden de Compra:** Comprende las solicitudes que una sucursal hace por una bodega a una proveedor para comprar uno o más productos. Esta entiende los diferentes productos a comprar, la cantidad, un precio acordado por unidad, una fecha esperada para su recepción, una fecha de llegada y una de creación de la orden. Toda orden corresponde a un proveedor.

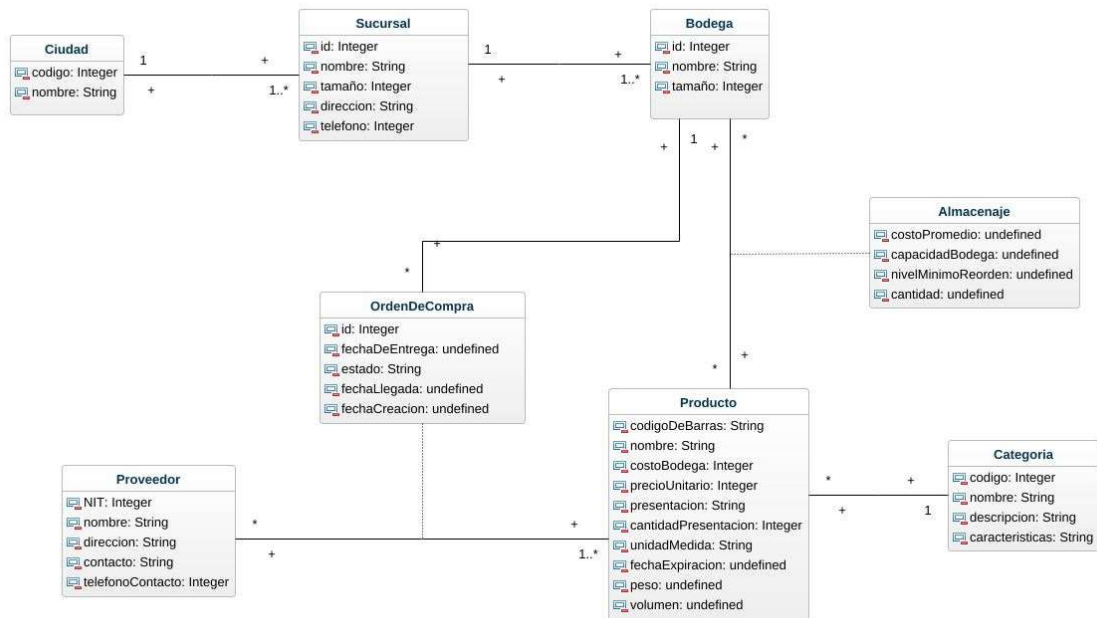
Nicolás Ballén-202310273
María Juliana Ballesteros-202313216
Mauricio Martinez-202314461

Modelo UML

Antiguo:



Nuevo:



Nicolás Ballén-202310273
María Juliana Ballesteros-202313216
Mauricio Martinez-202314461

Cambios Realizados:

- Recepción de Productos, deja de existir y su atributo (fechaLlegada) se integra a la entidad OrdenDeCompra, como se había dicho anteriormente en la entrega 1.
- A la entidad OrdenDeCompra se le integra otro atributo el cual es fechaCreacion.
- Se le añadió el atributo de cantidad del producto a la entidad Almacenaje.
- Se deshizo de especificEmpacado de la entidad Producto, para separar dichos valores en los atributos peso y volumen.

Modelo Relacional / Normalización

A continuación se presentarán las tablas/relación del modelo resultante, su llave candidata y por último un recorrido de como estas cumplen las formas normalizadas des 1FN a FN-BC.

Ciudades:

iD	Nombre
PK	NN,ND

Solo conserva un iD y nombre que no puede ser nulo y asumimos que no puede repetirse (en Colombia no hay ciudades con el mismo nombre)

- Llave candidata: iD
- 1FN: No tenemos atributos multivalor. Toda ciudad tiene un único id y nombre
- 2FN: La llave candidata es única, así que nombre depende de esta en su totalidad.
- 3FN: Como solo hay 1 atributo no primo (nombre), es imposible incumplir esta forma.
- BC: Hay una única llave candidata que es el único primo, así que no hay otro primo del que depender.

Sucursales:

iD	Nombre	Tamaño	Ciudad	Direccion	Telefono
PK	NN, ND	NN	NN, FK-Ciudades	NN	NN

Para cumplir la asociación uno a muchos de ciudad a sucursal, se toma un atributo ciudad en sucursales que no puede ser nulo, pero puede repetirse, de tal forma que una ciudad tenga varias sucursales asignadas a una única ciudad.

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

- Llave candidata: iD
- 1FN: No tenemos atributos multivalor. Todo atributo es atómico (no contemplamos múltiples teléfonos).
- 2FN: iD es el único primo, por lo que todos los demás atributos dependen en su totalidad de esta (sin parcialidad).
- 3FN: Ningún otro atributo puede determinar a los demás (solo iD), por lo que no podemos conocer ninguno de los atributos no primos sin el único primo (iD).
- BC: Hay una única llave candidata que es el único primo, así que no hay otro primo del que depender.

Bodegas

iD	Nombre	Tamaño	Sucursal
PK	NN	NN	NN, FK-Sucursales

Para cumplir con la asociación entre bodega y sucursal (muchos a uno), las bodegas tienen un atributo de sucursal que no puede ser nulo, más se puede repetir. De esta forma la bodega se asigna a una única sucursal, y una sucursal puede tener varias bodegas a su nombre.

- Llave candidata: iD
- 1FN: No tenemos atributos multivalor. Todo atributo es atómico.
- 2FN: iD es el único primo, por lo que todos los demás atributos dependen en su totalidad de esta (sin parcialidad).
- 3FN: Ningún otro atributo puede determinar a los demás (solo iD), por lo que no podemos conocer ninguno de los atributos no primos sin el único primo (iD).
- BC: Hay una única llave candidata que es el único primo, así que no hay otro primo del que depender.

Productos

Cod Barras	No mbr e	Costo en Bodega	Precio Unitario	Presen tación	CantidadPre sentacion	Unidad Medida	Peso	Volumen	Fecha de Expiracion	Categ oría
PK	NN	NN	NN	NN	NN	NN, CK	NN	NN	NN	NN, FK-Categ orias

En el caso de producto, cumplimos su relación muchos a uno con categoría con un atributo de categoría que no puede ser nulo pero si repetible, habiendo así varias productos con una única categoría, la cual puede tener varias productos.

- Llave candidata: Código de Barras
- 1FN: No tenemos atributos multivalor. Todo atributo es atómico.
- 2FN: Código de Barras es el único primo, por lo que todos los demás atributos dependen en su totalidad de esta (sin parcialidad).

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

- 3FN: Ningún otro atributo puede determinar a los demás (solo el código), por lo que no podemos conocer ninguno de los atributos no primos sin el único primo.
- BC: Hay una única llave candidata que es el único primo, así que no hay otro primo del que depender.

Almacenes

Bodega	Producto	Capacidad	Costo Promedio	Nivel Mínimo
PK, FK-Bodegas	PK, FK-Productos	NN	NN	NN

Esta relación sirve para comprender los muchos productos que pueden guardarse en múltiples bodegas, teniendo así tanto la llave de bodega como la de producto

- Llave candidata: (Bodega, Producto)
- 1FN: No tenemos atributos multivalor. Todo atributo es atómico.
- 2FN: En este caso se necesita si o se tanto la bodega como el producto para definir los demás atributos. Dos bodegas pueden tener el mismo producto, pero la capacidad, costo, y nivel mínimo que tienen en cada pueden ser distintos. De igual forma una bodega tendrán varios productos, pero los atributos de estos dentro de ella no serán necesariamente iguales. Por ello se necesitan ambos para ser si o si la llave candidata.
- 3FN: Los datos ajenos a la llave candidata corresponden únicamente al producto en la bodega correspondiente, por lo que estos no pueden determinarse entre sí. No hay no primos que determinen no primos.
- BC: La llave candidata es única, así como bodega no puede determinar el producto en cuestión (la bodega contiene muchos productos) ni viceversa (el producto se puede guardar en muchas bodegas). Por ende ningún primo determina otro primo.

Proveedor

NIT	Nombre	Contacto	Tel. Contacto
PK	NN	NN	NN

- Llave candidata: NIT
- 1FN: No tenemos atributos multivalor. Todo atributo es atómico.
- 2FN: El NIT es la llave candidata en su totalidad, y al ser la única significa que el resto de atributos (no primos) son determinados en su totalidad por esta.
- 3FN: El nombre no determina el contacto ni su teléfono. Podría existir que el número de teléfono no se repitiese y con este se determinará el resto, más este no es el caso.
- BC: La llave candidata es única, así como el único primo. Por ende no hay dependencia alguna entre primos.

Categorías

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

Codigo	Nombre	Descripción	Características
PK	NN, ND	NN	NN

- Llave candidata: Código
- 1FN: No tenemos atributos multivalor. Todo atributo es atómico (las descripciones y características son un único texto, seguramente de longitud mayor).
- 2FN: El codigo es la llave candidata en su totalidad, y al ser la única significa que el resto de atributos (no primos) son determinados en su totalidad por esta.
- 3FN: El resto de atributos no se determinan entre si.
- BC: La llave candidata es única, así como el único primo. Por ende no hay dependencia alguna entre primos.

Ofertas

Proveedor	Producto
PK, FK-Bodegas	PK, FK-Productos

Esta relación sirve a la asociación muchos a muchos entre proveedores y productos, definiendo los productos que ofrecen los diferentes proveedores.

- Llave candidata: (Proveedor, Producto)
- 1FN: No tenemos atributos multivalor. Todo atributo es atómico (se usan las llaves primarias de cada uno).
- 2FN: No hay atributos no-primos como tal, así que no puede haber dependencias.
- 3FN: No hay atributos no-primos como tal, así que no puede haber dependencias entre estos
- BC: Los dos atributos componen la única llave candidata, así que no pueden determinarse entre sí. De por sí solo se tiene el dato de ambos, los cuales se repiten para los múltiples proveedores con múltiples productos.

Ordenes

iD	Proveedor	FechaEstimada	FechaLlegada	FechaCreacion	Estado
PK	NN, FK-Proveedores	NN	NN	NN	NN, CK

En este caso destaco la relación muchos a uno de orden a proveedor, teniendo la orden un atributo repetible Proveedor que hace referencia al mismo en múltiples órdenes. Adicionalmente, su estado se marca con CK dados los determinados estados que puede tener la orden desde VIGENTE hasta ENTREGADA.

- Llave candidata: iD
- 1FN: No tenemos atributos multivalor. Todo atributo es atómico (se usan las llaves primarias en la foránea).
- 2FN: Como la llave candidata es atómica, si o si todos los demás atributos dependen de esta en su totalidad.

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

- 3FN: Ningún otro atributo que no sea el iD puede determinar al resto.
- BC: La llave candidata, así como los atributos primos, son solo uno. Ningún primo determina a otro.

ProductosOrden

Orden	Producto	Cantidad	Precio Acordado
PK, FK-Orden	PK, FK-Productos	NN	NN

Sirve a la relación muchos a muchos de orden y productos.

- Llave candidata: (Orden, Producto)
- 1FN: No tenemos atributos multivalor. Todo atributo es atómico (se usan las llaves primarias en la foránea).
- 2FN: Necesitamos tanto de la orden como del producto para saber cuanto del producto se ordenó y cual fue el precio acordado por unidad. Sin producto no sabemos de qué producto en específico hablamos (la orden tiene puede tener muchos), y sin orden no sabremos cuanto del producto se pidió y por cuanto (dependiendo del proveedor en la orden los datos serán diferentes). Por ende, se necesita la llave candidata en su totalidad.
- 3FN: La cantidad del producto no define el precio acordado, sino el producto y la orden. Ningún no primo define otro no primo.
- BC: La llave candidata es única. Ningún primo determina a otro (la orden no define al producto en específico ni le producto a la orden)

Escenarios de prueba

NOTA: Datos de las pruebas de postman pueden varias con respecto a los de estos escenarios

RF 1. CREAR UNA CIUDAD

Ciudades

id	Nombre
PK	NN,ND
1234	Bogotá
3456	Bucaramanga

1. Pruebas de unicidad de tuplas

Caso 1: Crear una nueva ciudad con un código nuevo

Se inserta una ciudad con un id **“4566”** y nombre **“Medellín”**. La operación debe ser exitosa ya que, la PK no se encuentra registrada.

La ciudad se agrega con éxito ya que no existe otra ciudad con el mismo id.

Caso 2: Crear una ciudad con un código que ya existe

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

Se inserta una ciudad con el id **"1234"** que ya se encuentra registrado y nombre **"Cali"**. La operación debe fallar ya que, no cumple con la restricción de unicidad de la PK.

2. Pruebas de integridad con FK

En este caso ciudad no depende de una FK.

3. Pruebas de integridad de acuerdo con restricciones de chequeo

Caso 1: Crear una ciudad sin nombre

Se intenta crear una ciudad sin nombre. Por ejemplo se inserta la ciudad con id **"7890"** y nombre **NULL**. Esta operación falla debido a que el atributo "nombre" de ciudad es NN.

Caso 2: Crear ciudad con el nombre repetido

Si se intenta crear una ciudad con un nombre que ya se encuentra en la tabla la operación va a fallar debido a que "Nombre" es ND.

Por ejemplo, se inserta la ciudad con id **"2211"** y nombre **"Bucaramanga"**. No se va a agregar la ciudad ya que se encuentra en la tabla.

RF 2. CREAR UNA SUCURSAL

Sucursales

id	Nombre	Tamaño	Ciudad	Dirección	Telefono
PK	NN,ND	NN	NN,FK-Ciudades	NN	NN
2024	Sucursal 1	20m ²	123	Calle 127	33445

1. Pruebas de unicidad de tuplas

Caso 1: Crear una sucursal nueva con un ID (PK) nuevo

Inserta una sucursal nueva con id **"2023"**, nombre **"Sucursal 2"**, tamaño **"50m²"**, dirección **"Calle 134"** y teléfono **"1111"** y ciudad **"1234"**. La operación debe ser exitosa ya que el id **"2023"** no se encuentra registrado.

Caso 2: Crear una sucursal con un ID (PK) existente

Se intenta crear una sucursal con el id **"2024"**, nombre **"Sucursal 3"**, tamaño **"50m²"**, dirección **"Calle 146"** y teléfono **"10101"** y ciudad **"3456"**. La operación debe fallar ya que, el id **"2024"** ya existe y no estaría cumpliendo con la restricción de unicidad de la PK.

2. Pruebas de integridad con FK

Caso 1: Crear una sucursal en una ciudad que se encuentra registrada

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

Se crea una sucursal en la ciudad **"1234"** (FK), la ciudad ya se encuentra registrada en la tabla de ciudades entonces se crea sin problema.

Caso 2: Crear una sucursal en una ciudad que no esta registrada

Se intenta crear una sucursal en la ciudad **"1321"** (FK). La operación debe fallar debido a que no se puede crear una sucursal con una ciudad que no existe.

3. Pruebas de integridad de acuerdo con restricciones de chequeo

Caso 1: Crear una sucursal sin nombre, tamaño, dirección y/o teléfono.

Si se intenta crear una sucursal sin nombre, tamaño, dirección y/o teléfono, la operación debe fallar debido a que estos atributos están definidos con una restricción NN. Por ejemplo, se inserta la sucursal con id **"4563"**, nombre **"Sucursal 10"**, tamaño **NULL**, dirección **"Calle 67"** y teléfono **"89007"**. Esta operación falla ya que no se aceptan valores nulos.

RF 3. CREAR Y BORRAR UNA BODEGA

Bodegas

id	Nombre	Tamaño	Sucursal
PK	NN	NN	NN, FK-Sucursales
2005	Bodega 1	30m ²	2024
3048	Bodega 2	30m ²	2024

1. Pruebas de unicidad de tuplas

Caso 1: Crear una bodega nueva con un ID (PK) nuevo

Se crea una bodega nueva y se le asigna el id **"2003"**, nombre **"Bodega 3"**, Tamaño **"40m²"** y una FK que apunta a la sucursal **"2024"**. La operación es exitosa ya que cumple con la restricción de unicidad de la PK.

Caso 2: Crear una bodega nueva con un ID (PK) existente

Se intenta crear una bodega nueva y se le asigna el id **"2005"**, nombre **"Bodega 3"**, Tamaño **"40m²"** y una FK que apunta a la sucursal **"2024"**, la operación falla ya que no cumple con la restricción de unicidad de la PK.

Caso 3: Borrar una bodega con ID (PK) existente

Se borra una bodega con un el id **"2005"** el cual ya existe en la tabla. La operación se ejecuta sin problema ya que la bodega existe.

Caso 4: Borrar una bodega con un ID inexistente

Se intenta borrar una bodega con el id **"1234"** el cual no esta registrado en la tabla. La operación falla ya que no se puede borrar una bodega que no existe.

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

2. Pruebas de integridad con FK

Caso 1: Crear una bodega asociada a una sucursal existente

Crea una bodega con una FK que apunta a el id **“2024”** de una sucursal. La creación debe ser exitosa ya que, la sucursal con el id **“2024”** se encuentra registrada en la tabla de sucursales.

Caso 2: Crear una bodega asociada a una sucursal inexistente

Se intenta crea una bodega con una FK que apunta al id **“2023”** de una sucursal de la tabla (la sucursal no existe). La bodega no se debe crear ya que no se puede crear una bodega sin una sucursal.

Caso 3: Borrar una bodega con productos

Almacenes

Bodega	Producto	Capacidad	Costo promedio	Nivel mínimo	Cantidad
PK, FK-Bodegas	PK, FK-Productos	NN	NN	NN	NN
2005	f0f0f0f0f0	10m ²	5.000 COP	20	10

Si se intenta borrar una bodega referenciada en “Almacenaje”, esta operación debe fallar, ya que, si se encuentra ahí significa que tiene productos y los productos se borrarían. Por ejemplo, se intenta borrar la bodega con id **“2005”**. La operación falla porque si se borra, también se borrarían los productos.

Caso 4: Borrar una bodega sin productos

Se borra una bodega que no se encuentre referenciada en “Almacenaje”. La bodega se borra con éxito ya que no se estarían eliminando productos. Por ejemplo, se desea borrar la bodega con id **“3048”**. La operación se realiza con éxito ya que la bodega no tiene productos asociados y se encuentra en la tabla de bodegas.

3. Pruebas de integridad de acuerdo con restricciones de chequeo

Caso 1: Crear una bodega sin nombre, tamaño y/o sucursal

Si se intenta crear una bodega sin nombre y/o tamaño, la operación falla, ya que estos atributos son NN. Por ejemplo se intenta crear una bodega con id **“4873”**, nombre **“Sucursal 3”**, tamaño **NULL** y proveedor y sucursal **“2024”**. La bodega no se debe crear ya que debe tener un tamaño porque la restricción de este campo es NN.

RF 4. CREAR Y ACTUALIZAR PROVEEDORES

Proveedores

Nicolás Ballén-202310273
María Juliana Ballesteros-202313216
Mauricio Martinez-202314461

NIT	Nombre	Contacto	Tel. Contacto	Dirección
PK	NN	NN	NN	NN
101108106	Proveedor 1	Ana	314 5678912	Cra. 15 #93-42

1. Pruebas de unicidad de tuplas

Caso 1: Crear un proveedor con un NIT único

Se crea un proveedor con un NIT (PK) único. La operación es exitosa ya que se cumple con la restricción de unicidad de la PK. Por ejemplo, se crea el proveedor con NIT **“194783984”**, nombre **“Proveedor 2”**, contacto **“Juan”** y teléfono de contacto **“3026577899”**. El proveedor se crea sin problema.

Caso 2: Crear un proveedor con un NIT existente

Si se intenta crear un proveedor con un NIT (PK) que ya se encuentra en la tabla de proveedores, la operación debe fallar ya que no cumple con la restricción de unicidad de la PK. Por ejemplo, se crea el proveedor con NIT **“101108106”**, nombre **“Proveedor 2”**, contacto **“Juan”** y teléfono de contacto **“3026577899”**. El proveedor no se crea ya que la PK ya está registrada.

2. Pruebas de integridad con FK

En este caso proveedor no depende de una FK.

3. Pruebas de integridad de acuerdo con restricciones de chequeo

Caso 1: Crear un proveedor sin los datos de su nombre, contacto y teléfono de contacto.

Si se intenta crear un proveedor con alguno de sus atributos como NULL la operación falla debido a la restricción de NN. Por ejemplo, se intenta crear el proveedor con NIT **“194783984”**, nombre **NULL**, contacto **“Juan”** y teléfono de contacto **“3026577899”**. El proveedor no se crea ya que, debe tener un nombre.

RF 5. CREAR Y LEER UNA CATEGORIA DE PRODUCTO

Categorías

Codigo	Nombre	Descripcion	Caracteristicas
PK	NN, ND	NN	NN
10100	Perecedero	Tiene fecha de vencimiento	Se deben consumir antes de la fecha de expiración

1. Pruebas de unicidad de tuplas

Caso 1: Crear una categoría con un código (PK) único

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

Se crea una categoría con código **“00101”**, nombre **“Congelados”**, descripción **“Se mantienen congelados”**, características **“Se deben mantener congelados”**. La categoría se crea sin problema ya que, no se encuentra registrada en la tabla.

Caso 2: Crear una categoría con un código (PK) duplicado

Se intenta crear la categoría con código **“10100”**, nombre **“Congelados”**, descripción **“Se mantienen congelados”**, características **“Se deben mantener congelados”**. La categoría no se crea ya que, no cumple con la restricción de unicidad de la PK.

2. Pruebas de integridad con FK

En este caso categoría no depende de una FK.

3. Pruebas de integridad de acuerdo con restricciones de chequeo

Caso 1: Crear una categoría con un nombre duplicado

Se intenta crear la categoría con código **“9348”**, nombre **“Perecedero”**, descripción **“Tiene fecha de vencimiento”** y características **“Se deben consumir antes de la fecha de expiración”**. La categoría no se crea ya que, ya existe una categoría con el mismo nombre y se incumpliría la restricción de ND de “nombre”.

Caso 2: Crear una categoría sin nombre, descripción y características.

Se crea una categoría con código **“00101”**, nombre **“Congelados”**, descripción **NULL**, características **“Se deben mantener congelados”**. La operación falla debido a que una categoría debe tener una descripción.

RF 6. CREAR, LEER Y ACTUALIZAR UN PRODUCTO

codB arras	no mbr e	costoB odega	precioU nitario	Presen tación	Cantid adPres	Unidad Medida	Pe so	Volu men	Fecha Exp	Cate goria
PK	NN ,	NN	NN	NN	NN	NN,CK	N N	NN	NN	FK- Cate goria
f0f0f 0f0f0	Pap as frita s	\$4200	\$8000	paquet ón de 5 paquet es de 200 gr. cada uno	1000 gr	gr	12 gr	150 cm ³	03/08 /2027	1010 0

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

1. Pruebas de unicidad de tuplas

Caso 1: Crear un producto con código de barras (PK) único

Se crea un producto con código de barras **"A12345B"**. El producto se crea sin problema ya que, el código de barras no se encuentra registrado.

Caso 2: Crear un producto con código de barras ya existente

Se crea un producto con código de barras **"f0f0f0f0f0"**. El producto no ya que, el código de barras ya se encuentra registrado para otro producto.

2. Pruebas de integridad con FK

Caso 1: Crear un producto con una categoría (FK) valida

Se crea un producto con código de barras **"01929392"**, y categoría **"10100"**. El producto se crea sin problema ya que la FK que referencia a una categoría existe en la tabla de categorías.

Caso 2: Crear un producto con una categoría (FK) que no existe

Se intenta crear un producto con código de barras **"01929392"**, y categoría **"11111"**. El producto no se crea ya que la categoría no existe y se incumpliría con la integridad referencial.

3. Pruebas de integridad de acuerdo con restricciones de chequeo

Caso 1: Insertar un producto con unidades de medida incorrectas

Se intenta insertar un producto con unidades de medida **"Kg"**. El producto no se inserta ya que, la restricción de chequeo indica que un producto debe tener unidades de gr o ml.

4. Prueba de lectura producto

Caso 1: Leer un producto usando el código de barras existente

Se desea leer el producto con código de barras **"f0f0f0f0f0"**. La operación se ejecuta sin problemas ya que el producto se encuentra registrado

Caso 2: Leer un producto usando un código de barras que no existe

Se intenta leer el producto con código de barras **"0000000000"**. La lectura del producto falla debido a que este no se encuentra registrado en la tabla de productos.

5. Prueba de actualización producto

Caso 1: Actualizar un producto usando el código existente y actualizando información valida

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

Se intenta actualizar un producto con el código “f0f0f0f0f0” y la información que se actualiza es la descripción “Consumir después de abierto”. La información se actualiza sin problema.

Caso 2: Actualizar un producto usando información invalida

Se intenta actualizar la categoría de un producto por una que no existe. Por ejemplo, quiero actualizar el producto “f0f0f0f0f0” y la información que actualizo es la categoría “93847”, la información no se puede actualizar ya que esta categoría no existe.

RF 7. CREAR UNA ORDEN DE COMPRA PARA UNA SUCURSAL

Ordenes

id	proveedor	fechaEstimada	fechaLlegada	fechaCreacion	Estado
PK	FK-Proveedores	NN	NN	NN	NN,CK
1	101108106	28/12/2024	29/12/2024	21/12/2024	Vigente
2	101108106	12/12/2024	14/12/2024	10/12/2024	Entregada

1. Pruebas de unicidad de tuplas

Caso 1: Crear una orden de compra con un id (PK) único

Se crea una orden de compra con id “2”. La orden se crea con éxito ya que no existe una con el mismo id.

Caso 2: Crear una orden de compra con un id (PK) duplicado

Se intenta crear una orden de compra con id “1”. La orden de compra no se crea debido a que ya existe una con el mismo id.

2. Pruebas de integridad con una FK

Caso 1: Crear una orden de compra con un proveedor existente

Se crea una orden de compra con id “3”, proveedor “101108106”, fecha estimada “28/12/2024”, estado “Vigente”. La orden se crea con éxito debido a que el proveedor existe en la tabla de proveedores.

Caso 2: Crear una orden de compra con un proveedor que no existe

Se intenta crear una orden de compra con id “3”, proveedor “109874783”, fecha estimada “28/12/2024”, estado “Vigente”. La orden no se crea con éxito debido a que el proveedor no existe en la tabla de proveedores y se incumpliría con la integridad de referencial.

Caso 3: crear una orden de compra con una Bodega que existe

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

RF 8. ACTUALIZAR UNA ORDEN DE COMPRA CAMBIANDO SU ESTADO A ANULADA

id	proveedor	fechaEstimada	fechaLlegada	fechaCreacion	Estado
PK	FK-Proveedores	NN	NN	NN	NN,CK
1	101108106	28/12/2024	29/12/2024	21/12/2024	Vigente
2	101108106	12/12/2024	14/12/2024	10/12/2024	Entregada

1. Pruebas de unicidad de tuplas

Caso 1: Actualizar una orden de compra con un id (PK) único

Se actualiza el estado de una orden de compra con id **"1"**. La orden se actualiza con éxito ya que se encuentra registrada en la tabla de ordenes.

Caso 2: Actualizar una orden de compra con un id (PK) que no existe

Se intenta actualizar una orden de compra con el id **"3"**. La orden no se puede actualizar ya que no existe en la tabla.

2. Pruebas de integridad con una FK

Para actualizar una orden de compra no se necesita una FK. Solo se necesita el numero de la orden.

3. Pruebas de integridad de acuerdo con restricciones de chequeo

Caso 1: Si el estado actual de la orden de compra en **"Entregada"** no puede ser anulada

Si se intenta actualizar la orden de compra con id **"2"** y estado **"Anulada"** la operación va a fallar ya que, el estado actual de la orden de compra es **"entregada"** y la restricción de chequeo indica que una orden con estado **"Entregada"** no se puede cambiar por **"Anulada"**.

Caso 2: El estado de la orden de compra debe ser **"Vigente"** para poder ser anulada

Se intenta actualizar la orden de compra con id **"1"** y estado **"Anulada"**. La operación debe ser exitosa ya que, el estado actual de la orden de compra es **"Vigente"** y la restricción de chequeo solo permite actualizar el estado de **"Vigente"** a **"Anulada"**

RF 9. MOSTRAR TODAS LAS ORDENES DE COMPRA

Ordenes

id	proveedor	fechaEstimada	fechaLlegada	fechaCreacion	Estado
PK	FK-Proveedores	NN	NN	NN	NN,CK
1	101108106	28/12/2024	29/12/2024	21/12/2024	Vigente
2	101108106	12/12/2024	14/12/2024	10/12/2024	Entregada

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

1. Pruebas de unicidad de tuplas

En este caso no aplican directamente ya que, el requerimiento no se trata de insertar o actualizar una tupla. Para esto ya se debió verificar la unicidad de PK en los requerimientos anteriores.

2. Pruebas de integridad con una FK

Caso 1: Verificar que cuando se despliegue la lista de ordenes de compra las FK (proveedor y la bodega)

Cuando se despliegue la lista de ordenes el proveedor y bodega deben ser FK que apunten a un dato valido, en este caso que existe en las tablas.

3. Pruebas de integridad de acuerdo con restricciones de chequeo

Cuando se desplieguen las ordenes de compra en estado debe ser **“Vigente”**, **“Entregada”** o **“Anulada”**. Ya que la restricción de chequeo indica que el estado solo puede tomar uno de estos valores.

RFC1. MOSTRAR EL ÍNDICE DE OCUPACIÓN DE CADA UNA DE BODEGAS DE UNA SUCURSAL

IndiceOcupacion

Producto	VolumenOcupado	CapacidadBodega	Índice
FK- Productos	NN	NN	NN
f0f0f0f0f0	20	150	20

1. Pruebas de unicidad de tuplas

En este caso no aplican directamente ya que, el requerimiento no se trata de insertar o actualizar una tupla. Para esto ya se debió verificar la unicidad de PK en los requerimientos anteriores.

2. Pruebas de integridad con una FK

Verificar que cuando se despliegue la lista de los productos de cada uno de los productos de la bodega sea FK (productos)

Cuando se despliegue la lista de los productos, cada producto de la lista debe ser un FK que apunte a un dato valido, en este caso que existe en las tablas.

3. Pruebas de integridad de acuerdo con restricciones de chequeo

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

El índice de ocupación no puede dar negativo.

RFC2. MOSTRAR LOS PRODUCTOS QUE CUMPLEN CON CIERTA CARACTERÍSTICA

ProductoCaracteristica

Producto	Precio	FechaVencimiento	Sucursal	Categoria
PK, FK- Productos	NN	NN	NN, FK - Sucursal	NN, FK - Categoria
f0f0f0f0f0	11.300 COP	12/12/2023	2022	10101

1. Pruebas de unicidad de tuplas

En este caso no aplican directamente ya que, el requerimiento no se trata de insertar o actualizar una tupla. Para esto ya se debió verificar la unicidad de PK en los requerimientos anteriores.

2. Pruebas de integridad con una FK

Verificar que cuando se despliegue la lista de los productos de cada uno de los productos de la bodega sea FK (productos, sucursal y categoría).

Cuando se despliegue la lista de los productos, cada producto, sucursal y categoría de la lista debe ser un FK que apunte a un dato valido, en este caso que existe en las tablas.

3. Pruebas de integridad de acuerdo con restricciones de chequeo

Ya fueron previamente revisadas en los requerimientos anteriores.

RFC3. INVENTARIO DE PRODUCTOS EN UNA BODEGA

InventarioBodega

Producto	CantidadActual	CantidadMinima	CostoPromedio
PK, FK- Productos	NN	NN	NN
f0f0f0f0f0	100	25	42.300 COP

1. Pruebas de unicidad de tuplas

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

En este caso no aplican directamente ya que, el requerimiento no se trata de insertar o actualizar una tupla. Para esto ya se debió verificar la unicidad de PK en los requerimientos anteriores.

2. Pruebas de integridad con una FK

Verificar que cuando se despliegue la lista de almacenaje de cada uno de los productos de la bodega de compra las FK (productos)

Cuando se despliegue la lista de almacenaje, cada producto de la lista debe ser un FK que apunte a un dato valido, en este caso que existe en las tablas.

3. Pruebas de integridad de acuerdo con restricciones de chequeo

Ya fueron previamente revisadas en los requerimientos anteriores.

RFC4. MOSTRAR LAS SUCURSALES EN LAS QUE HAY DISPONIBILIDAD DE UN PRODUCTO

SucursalProducto

id	Nombre	Tamaño	Ciudad	Dirección	Telefono
PK	NN,ND	NN	NN,FK-Ciudades	NN	NN
2024	Sucursal 1	20m ²	123	Calle 127	33445

1. Pruebas de unicidad de tuplas

En este caso no aplican directamente ya que, el requerimiento no se trata de insertar o actualizar una tupla. Para esto ya se debió verificar la unicidad de PK en los requerimientos anteriores.

2. Pruebas de integridad con una FK

Verificar que cuando se despliegue la tabla de las sucursales, cada una de las ciudades tiene que ser FK (ciudades)

Cuando se despliegue la lista de sucursales, cada ciudad de cada sucursal de la tabla debe ser un FK que apunte a un dato valido, en este caso que existe en las tablas.

3. Pruebas de integridad de acuerdo con restricciones de chequeo

Ya fueron previamente revisadas en los requerimientos anteriores.

Nicolás Ballén-202310273
María Juliana Ballesteros-202313216
Mauricio Martinez-202314461

RFC5. MOSTRAR TODOS LOS PRODUCTOS QUE REQUIEREN UNA ORDEN DE COMPRA

ProductoNull

codB arras	no mbr e	costoB odega	precioU nitario	Presen tación	Cantid adPres	Unidad Medida	Pe so	Volu men	Fecha Exp	Cate goria
PK	NN ,	NN	NN	NN	NN	NN,CK	N N	NN	NN	FK- Cate goria
f0f0f 0f0f0	Pap as frita s	\$4200	\$8000	paquet ón de 5 paquet es de 200 gr. cada uno	1000 gr	gr	12 gr	150 cm ³	03/08 /2027	1010 0

1. Pruebas de unicidad de tuplas

En este caso no aplican directamente ya que, el requerimiento no se trata de insertar o actualizar una tupla. Para esto ya se debió verificar la unicidad de PK en los requerimientos anteriores.

2. Pruebas de integridad con una FK

No cuenta con ninguna, ya que la tabla resultante no cuenta con FK.

3. Pruebas de integridad de acuerdo con restricciones de chequeo

Nicolás Ballén-202310273

María Juliana Ballesteros-202313216

Mauricio Martinez-202314461

Ya fueron previamente revisadas en los requerimientos anteriores.

Resultados Pruebas Postman de Consultas:

NOTA: Por dificultades que no pudimos deducir (en vista de que la aplicación se ejecutaba), las pruebas no pudieron verificarse en postman al no poder conectar con ciertas tablas. Por ello, con tal de mostrar lo correcto de las sentencias SQL, a continuación se muestran resultados de los mismos desde SQL Developer. De igual forma se incluyeron dichos scripts

RC1-sucursal id 1

```
--RFC1
SELECT almacenajes.bodega b, almacenajes.producto p,
productos.volumen*almacenajes.cantidad/almacenajes.capacidad indice_ocupacion
FROM almacenajes INNER JOIN productos ON almacenajes.producto = productos.cod_barras
INNER JOIN bodegas on almacenajes.bodega = bodegas.id WHERE bodegas.sucursal = 1;
```

	B	P	INDICE_OCUPACION
1	7	2	6248,923401642489830378386675876890014583
2	7	3	9077,354182138878574080950612699591533606
3	1	8	2455,311089236617901632666915583313533146
4	1	2	3045,850605582435496849299321789472584202

RC2-2.1 a 2.4

```
--RFC2.1
SELECT * FROM productos WHERE precio_unitario BETWEEN 500 AND 5000;
--RFC2.2
SELECT * FROM productos WHERE fecha_vencimiento > TO_DATE('2024-01-01', 'YYYY-MM-DD');
--RFC2.3
SELECT * FROM productos WHERE fecha_vencimiento < TO_DATE('2024-01-01', 'YYYY-MM-DD');
--RFC2.4
```

	COD_BARRAS	NOMBRE	COSTO_BODEGA	PRECIO_UNITARIO	PRESENTACION
1	8	Alphazap	31147	894	Certotrichas paena
2	12	Sonair	1790	635	Hystrix indica
3	14	Voltzillam	46	2392	Choloepus hoffmani
4	15	Quo Lux	8	585	Phalacrocorax carbo
5	16	Flowdesk	3	2605	Felis silvestris lybica

Nicolás Ballén-202310273
 María Juliana Ballesteros-202313216
 Mauricio Martinez-202314461
RC3 inventario bodega 6

```
--RFC3
SELECT * FROM almacenajes WHERE almacenajes.bodega = 6;
--RFC4
```

	BODEGA	PRODUCTO	CAPACIDAD	CANTIDAD	COSTO_PROMEDIO	NIVEL_MINIMO
1	6	1	34164	6971	47930	21
2	6	8	79593	8863	31760	3262



RC4 sucursales con producto 1

```
--RFC4
SELECT * FROM sucursales INNER JOIN bodegas ON bodegas.sucursal=sucursales.id
INNER JOIN almacenajes on bodegas.id=almacenajes.bodega WHERE almacenajes.producto = 1;
```

ID	NOMBRE	TAMAÑO	CIUDAD	TELEFONO	DIRECCION	ID_1	NOMBRE_1	TAMAÑO_1	SUCURSAL	BODEGA	PRODUCTO
1	8 Shoshone	19549	9	156879693	11476 Bluestem Drive	8	Star Sedge	710	8	8	1
2	5 Redwing	19	1	721465833	6 Lien Junction	6	Fendler's Drymary	6	5	6	1
3	5 Redwing	19	1	721465833	6 Lien Junction	5	Lemmon's Poppy	2594	5	5	1

RC5 productos que requieren orden de compra

```
--RFC5
SELECT productos.cod_barras id, productos.nombre nombre,
almacenajes.bodega bodega, ofertas.proveedor proveedor, bodegas.sucursal sucursal,
almacenajes.cantidad cantidad FROM productos INNER JOIN almacenajes
ON productos.cod_barras=almacenajes.producto INNER JOIN bodegas
ON bodegas.id = almacenajes.bodega INNER JOIN ofertas ON productos.cod_barras=ofertas.producto
WHERE almacenajes.cantidad<almacenajes.nivel_minimo;
```

    SQL | Todas las Filas Recuperadas: 10 en 0,016 segundos

	ID	NOMBRE	BODEGA	PROVEEDOR	SUCURSAL	CANTIDAD
1	2	Zathin	1	9	1	3707
2	1	Bitwolf	5	2	5	8778
3	5	Wrapsafe	5	7	5	3730
4	2	Zathin	7	9	1	6213
5	1	Bitwolf	8	2	8	63
6	4	Zoolab	9	1	8	2417
7	6	Stim	9	1	8	7215
8	4	Zoolab	9	6	8	2417
9	5	Wrapsafe	9	7	8	426