UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA ESCUELA DE CIENCIAS Y
SISTEMAS SISTEMAS DE BASES DE DATOS 2
Seccion A
Erik Leonel Flores Pérez

Ventas S. A.

1. Modelo Conceptual

1.1 Diagrama Entidad-Relación

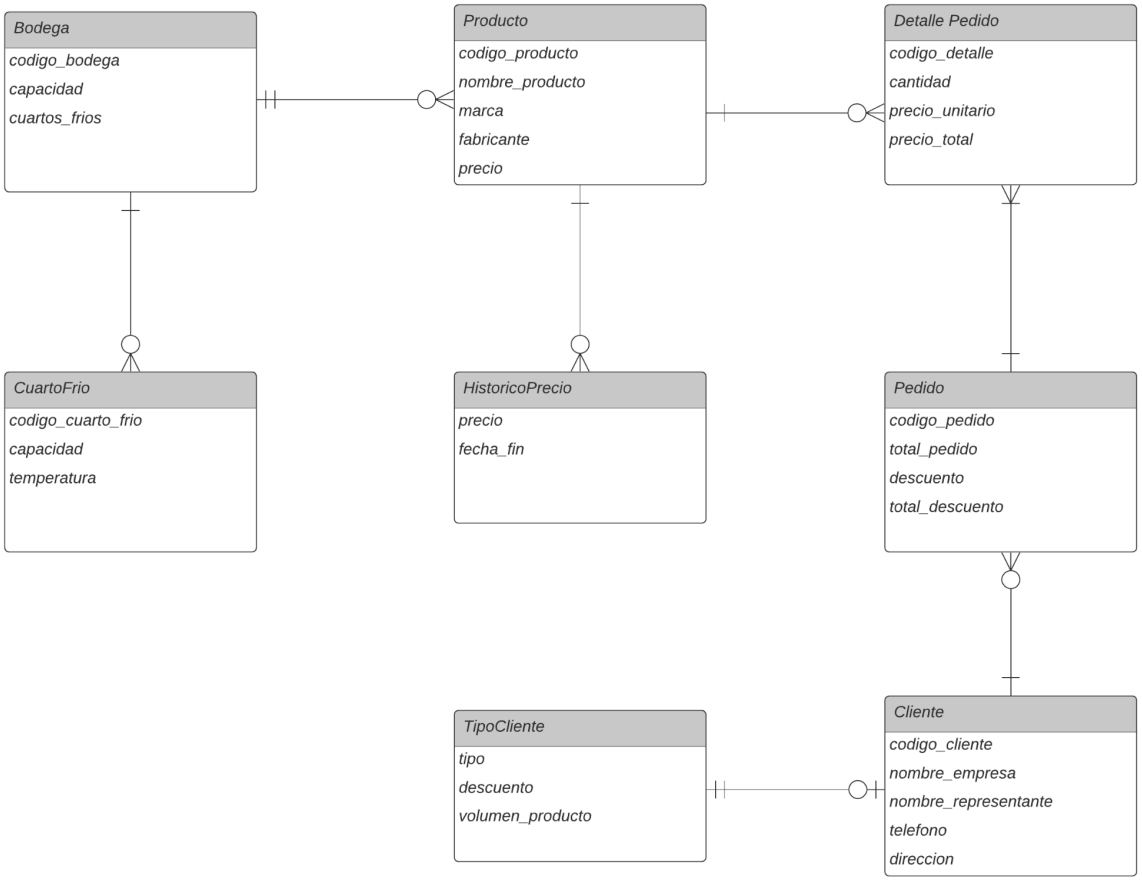


Diagrama 1.1.0 *Notación IDEF1X + Barker*

2. Modelo Lógico

2.1. Listado de Módulos

#	Módulo	Descripción
1	Bodegas	Gestiona las 34 bodegas de la empresa, incluyendo su capacidad en metros cúbicos y si tienen cuartos fríos para productos perecederos.
2	Clientes	Registra la información de los clientes, incluyendo su código, nombre de empresa, representante legal, tipo de empresa y condiciones de descuento.
3	CategoriaClientes	Se registra las categorias de los clientes, estos pueden reconfigurarselos .limites
4	CuartosFríos	Administra los cuartos fríos, incluyendo su código, capacidad en metros cúbicos y la temperatura constante para productos perecederos.
5	Productos	Gestiona los productos de la empresa, incluyendo su código, nombre, marca, fabricante, precio actual y su ubicación en las bodegas. Adicional el historico de los precios al ser modfiicados los productos.
6	Pedidos	Administra los pedidos realizados por los clientes, con detalle de productos, precios unitarios, y total por pedido, generando detalles para entrega, descuentos aplicados.
7	HistoricoProducto	Este modulo es interno, por lo que su función es registrar los cambios de precios de los productos, es una bitacora que puede ser consultada al cambiarse el precio de un producto.

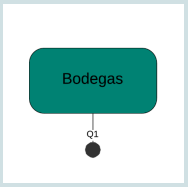
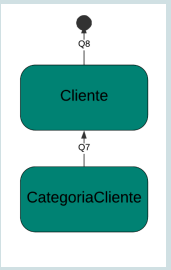
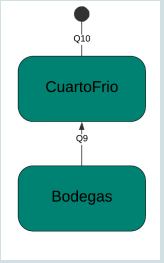
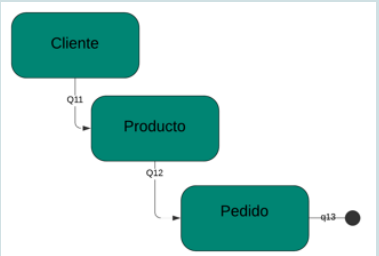
2.2. Modelado Queries

#	Módulo	Descripción
Q1	Bodegas	`SELECT bodega.codigo_bodega, bodega.capacidad AS capacidad_bodega, bodega.cuartos_frios FROM bodega WHERE bodega.codigo_bodega = :p;`
Q2	Productos	`SELECT bodega.codigo_bodega FROM bodega WHERE bodega.codigo_bodega = :p;`
Q3	Productos	`-- Seleccionar los detalles del producto y la bodega donde está almacenado SELECT producto.codigo_producto, producto.nombre_producto, producto.marca, producto.fabricante, producto.precio, bodega.codigo_bodega FROM producto, bodega WHERE producto.codigo_bodega = bodega.codigo_bodega;`
Q4	HistoricoProducto	`-- Seleccionar los detalles del producto y la bodega donde está almacenado SELECT producto.codigo_producto, producto.precio FROM producto WHERE producto.codigo_bodega =:p;`
Q5	HistoricoProducto	`-- Seleccionar el historial de precios de los productos, incluyendo el nombre del producto y la fecha de fin de precio SELECT producto.codigo_producto, producto.nombre_producto, historicoprecio.precio, historicoprecio.fecha_fin FROM historicoprecio, producto WHERE historicoprecio.codigo_producto =`

		producto.codigo_producto;
Q6	CategoríaCliente	SELECT tipocliente.tipo, tipocliente.descuento, tipocliente.volumen_producto FROM tipocliente WHERE tipocliente.tipo = :p ;
Q7	Clientes	SELECT tipocliente.tipo, tipocliente.descuento, tipocliente.volumen_producto FROM tipocliente WHERE tipocliente.tipo = :p ;
Q8	Clientes	SELECT cliente.codigo_cliente, cliente.nombre_empresa, cliente.nombre_representante, cliente.telefono, cliente.direccion, tipocliente.tipo, tipocliente.descuento, tipocliente.volumen_producto FROM cliente, tipocliente WHERE cliente.tipocliente = tipocliente.tipo AND cliente.codigo_cliente = :p;
Q9	CuartosFríos	SELECT bodega.codigo_bodega FROM bodega WHERE bodega.codigo_bodega = :p;
Q10	CuartosFríos	SELECT cuartofrio.codigo_cuarto_frio, cuartofrio.capacidad, cuartofrio.temperatura, bodega.codigo_bodega, FROM cuartofrio, bodega WHERE cuartofrio.codigo_cuarto_frio = bodega.codigo_bodega;
Q11	Pedidos y Detalle Pedidos	SELECT cliente.codigo_cliente, cliente.nombre_empresa, tipocliente.tipo, tipocliente.descuento, tipocliente.volumen_producto

		FROM cliente, tipocliente WHERE cliente.cliente = :p AND cliente.tipocliente = tipocliente.tipo;
Q12	Pedidos y Detalle Pedidos	SELECT producto.codigo_producto, producto.nombre_producto, producto.marca, producto.precio, bodega.codigo_bodega FROM producto, bodega WHERE producto.codigo_producto = :p AND producto.codigo_bodega = bodega.codigo_bodega;
Q13	Pedidos y Detalle Pedidos	-- Seleccionar todos los pedidos junto con su detalle de productos, cliente, y los descuentos aplicados SELECT pedido.codigo_pedido, pedido.total_pedido, pedido.descuento, pedido.total_descuento, cliente.codigo_cliente, cliente.nombre_empresa, detallepedido.cantidad, detallepedido.precio_unitario, detallepedido.precio_total, producto.nombre_producto bodega.codigo_bodega FROM pedido, detallepedido, cliente, producto, bodega WHERE pedido.codigo_pedido = detallepedido.codigo_pedido AND pedido.codigo_cliente = cliente.codigo_cliente AND detallepedido.codigo_producto = producto.codigo_producto AND producto.codigo_bodega = bodega.codigo_bodega;

2.3. Flujo Módulos

#	Módulo	Flujos
1	Bodegas	
2	Clientes	
3	CategoriaClientes	
4	CuartosFríos	
5	Productos	
6	Pedidos	

7	HistoricoProducto	

2.4. Diagrama Entidad-Relación

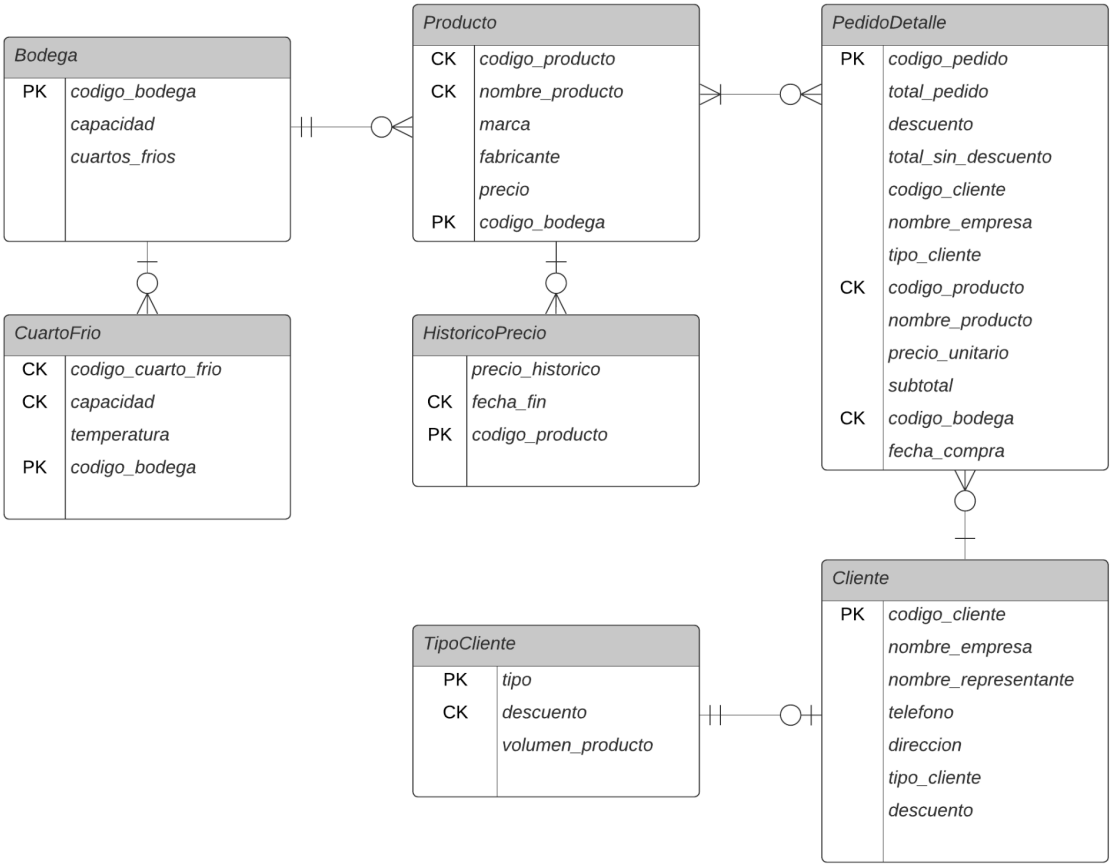


Diagrama 2.4.0 Diagrama Entidad Relación
Notación IDEF1X + Barker, **PK**: Partition Key/Primary Key, **CK**: Cluster Key

3. Modelo Físico

3.1. Diagrama Estructuras (Tablas)

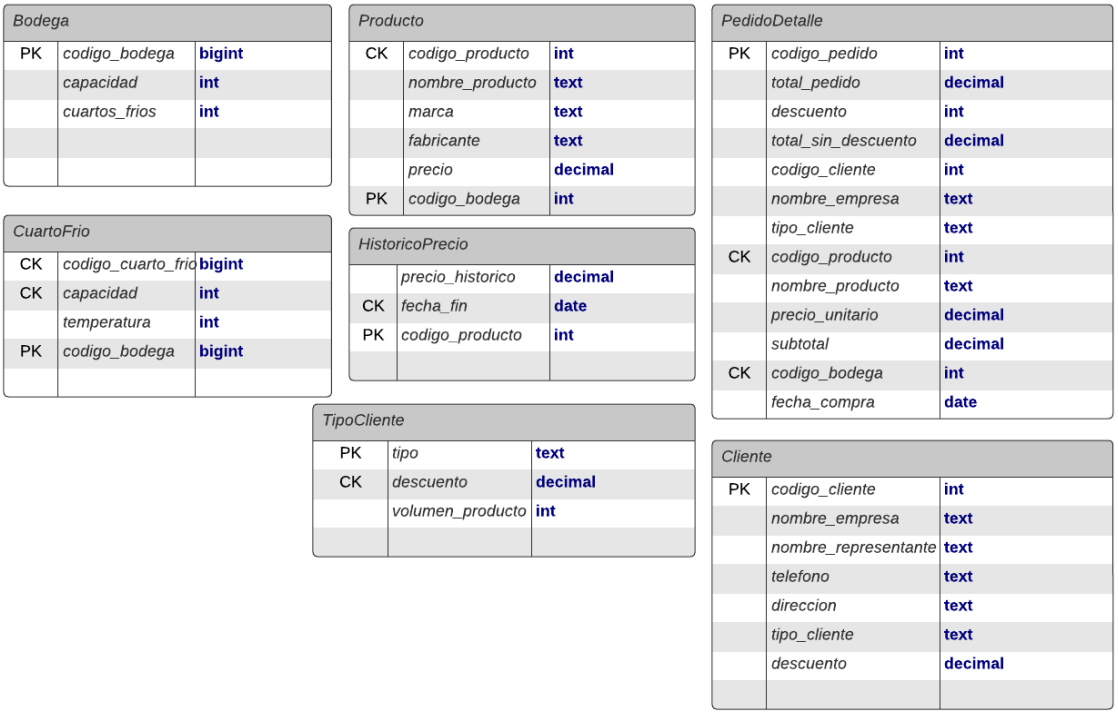


Diagrama 3.1.0 Diagrama Entidad Relación
PK: Partition Key/Primary Key, **CK**: Cluster Key

3.2. Particionamiento

1. Tabla Bodega

- Partition Key (PK): codigo_bodega
- Comentario: El uso de codigo_bodega como clave primaria es adecuado si los datos de cada bodega son únicos y accedidos frecuentemente en función de este identificador. Este esquema funcionará bien si las consultas se realizan por bodega.

2. Tabla CuartoFrio

- Partition Key (PK): codigo_bodega
- Clustering Keys (CK): codigo_cuarto_frio, capacidad
- Comentario: El particionamiento por codigo_bodega permite agrupar los cuartos fríos por bodega, lo cual es adecuado para consultas que recuperan todos los cuartos fríos de una bodega específica. Las claves de clúster adicionales (codigo_cuarto_frio y capacidad) ordenarán los resultados dentro de cada partición, lo cual es útil si se requieren búsquedas adicionales o si se ordenan los datos por estos campos.

3. Tabla Producto

- Partition Key (PK): codigo_bodega
- Clustering Key (CK): codigo_producto
- Comentario: El particionamiento por codigo_bodega es una buena opción para agrupar productos por bodega, lo que facilita las consultas que recuperan todos los productos de una bodega específica. La clave de clúster codigo_producto ayudará a ordenar los productos dentro de la partición, lo cual es útil si las consultas son frecuentes por codigo_producto.

4. Tabla HistoricoPrecio

- Partition Key (PK): codigo_producto
- Clustering Key (CK): fecha_fin
- Comentario: Esta combinación es adecuada si deseas consultar el historial de precios de un producto específico a lo largo del tiempo. El particionamiento por codigo_producto permite agrupar los registros por producto, mientras que la clave de clúster fecha_fin ayuda a ordenar los registros cronológicamente.

5. Tabla TipoCliente

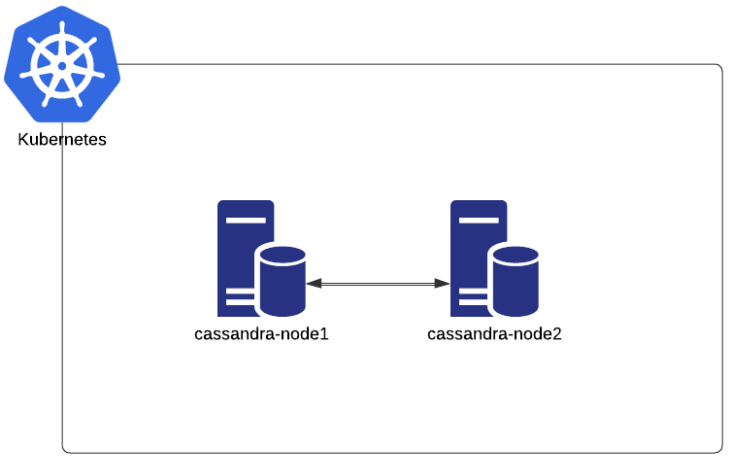
- Partition Key (PK): tipo
- Clustering Key (CK): descuento
- Comentario: Particionar por tipo puede ser adecuado si el número de tipos de clientes es limitado. Sin embargo, si hay pocos tipos de clientes y muchas filas por tipo, esto podría causar un desbalanceo en el almacenamiento. En ese caso, sería útil considerar una forma diferente de particionamiento o agregar un campo adicional a la clave de partición.

6. Tabla PedidoDetalle

- Partition Key (PK): codigo_pedido
- Clustering Keys (CK): codigo_cliente, codigo_producto, codigo_bodega, fecha_compra
- Comentario: El uso de codigo_pedido como clave de partición agrupa los detalles por pedido, lo cual es adecuado si se busca acceder a los detalles de un pedido en particular. Sin embargo, si hay muchos detalles en un solo pedido, puede ser necesario revisar el particionamiento para evitar particiones demasiado grandes. Las claves de clúster (codigo_cliente, codigo_producto, codigo_bodega, fecha_compra) permiten ordenar y realizar búsquedas detalladas dentro de la partición.

7. Tabla Cliente

- Partition Key (PK): codigo_cliente
- Comentario: El uso de codigo_cliente como clave primaria es adecuado para consultas que buscan un cliente específico. No se requieren claves de clúster adicionales en este caso.


Kubernetes
cassandra-node1
cassandra-node2

3.3. Scripts

```
CREATE TABLE Bodega (
	codigo_bodega bigint PRIMARY KEY,
	capacidad int,
	cuartos_frios int
);

CREATE TABLE CuartoFrio (
	codigo_cuarto_frio bigint,
	capacidad int,
	temperatura int,
	codigo_bodega bigint,
	PRIMARY KEY (codigo_bodega, codigo_cuarto_frio,
capacidad)
);

CREATE TABLE Producto (
	codigo_producto int,
	nombre_producto text,
	marca text,
	fabricante text,
	precio decimal,
	codigo_bodega int,
	PRIMARY KEY (codigo_bodega, codigo_producto)
);


CREATE TABLE HistoricoPrecio (
	codigo_producto int,
	precio_historico decimal,
	fecha date,
	PRIMARY KEY (codigo_producto, fecha)
);


CREATE TABLE TipoCliente (
	tipo text,
	descuento decimal,
	volumen_producto int,
	PRIMARY KEY (tipo, volumen_producto)
);


CREATE TABLE PedidoDetalle (
	codigo_pedido int,
	total_pedido decimal,
	descuento decimal,
	total_sin_descuento decimal,
	codigo_cliente int,
	nombre_empresa text,
	tipo_cliente text,
	codigo_producto int,
	nombre_producto text,
```

```
      precio_unitario decimal,
      subtotal decimal,
      codigo_bodega int,
      fecha_compra date,
      PRIMARY KEY (codigo_pedido, codigo_producto,
codigo_bodega)
);


CREATE TABLE Cliente (
      codigo_cliente int PRIMARY KEY,
      nombre_empresa text,
      nombre_representante text,
      telefono text,
      direccion text,
      tipo_cliente text,
      descuento decimal
);
```

3.4. Índices

En el diseño actual del esquema, se ha optado por no utilizar índices secundarios, principalmente debido a la incertidumbre en cuanto a las consultas comunes que se realizarán sobre las tablas de catálogo. Además, dado que la tabla transaccional (PedidoDetalle) no se utilizará frecuentemente para reportes, se ha priorizado un diseño optimizado para inserciones y consultas clave, en lugar de optimizar para búsquedas secundarias.

En resumen, este diseño es adecuado para:

- Operaciones de consulta directa sobre las claves primarias.
- Altos volúmenes de inserciones sin la carga adicional de mantenimiento de índices secundarios.
- Evitar consultas costosas que impliquen columnas no indexadas, dada la naturaleza desconocida de los patrones de consultas adicionales.

A medida que los patrones de consulta se vuelvan más claros, se puede considerar la creación de nuevas tablas diseñadas específicamente para reportes o el uso de índices secundarios para optimizar consultas específicas de columnas que no formen parte de las claves primarias, siempre y cuando las métricas de uso justifiquen su implementación.