-- Crear secuencias

CREATE SEQUENCE ROLE\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

CREATE SEQUENCE USER\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

CREATE SEQUENCE DETAIL\_CAR\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

CREATE SEQUENCE CAT\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

CREATE SEQUENCE DETAIL\_PED\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

CREATE SEQUENCE DIR\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

CREATE SEQUENCE IMG\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

CREATE SEQUENCE KARDEX\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

CREATE SEQUENCE MOV\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

CREATE SEQUENCE PED\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

CREATE SEQUENCE PROD\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

CREATE SEQUENCE ETIQUETA\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

CREATE SEQUENCE ETIQUETA\_PRODUCTO\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

CREATE SEQUENCE VAL\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

CREATE SEQUENCE PEDIDO\_PRODUCTO\_SEQ START WITH 1 INCREMENT BY 1 NOCACHE;

-- Tabla rol\_usuario

CREATE TABLE rol\_usuario (

id\_rol\_usuario INTEGER DEFAULT ROLE\_SEQ.NEXTVAL,

nombre\_rol VARCHAR2(20)

);

ALTER TABLE rol\_usuario ADD CONSTRAINT pk\_rol\_usuario PRIMARY KEY (id\_rol\_usuario);

-- Insertar roles en rol\_usuario

INSERT INTO rol\_usuario (id\_rol\_usuario, nombre\_rol) VALUES (ROLE\_SEQ.NEXTVAL, 'USUARIO');

INSERT INTO rol\_usuario (id\_rol\_usuario, nombre\_rol) VALUES (ROLE\_SEQ.NEXTVAL, 'ADMINISTRADOR');

-- Tabla usuario

CREATE TABLE usuario (

id\_user INTEGER DEFAULT USER\_SEQ.NEXTVAL,

email VARCHAR2(100),

password VARCHAR2(500),

activo NUMBER(1) DEFAULT 1, -- Definir como booleano

p\_nombre VARCHAR2(20),

m\_nombre VARCHAR2(20),

ap\_paterno VARCHAR2(20),

ap\_materno VARCHAR2(20),

telefono VARCHAR2(15),

role\_id INTEGER NOT NULL

);

ALTER TABLE usuario ADD CONSTRAINT pk\_usuario PRIMARY KEY (id\_user);

ALTER TABLE usuario ADD CONSTRAINT fk\_usuario\_rol\_usuario FOREIGN KEY (role\_id) REFERENCES rol\_usuario (id\_rol\_usuario);

-- Tabla categoria

CREATE TABLE categoria (

catid INTEGER DEFAULT CAT\_SEQ.NEXTVAL,

catname VARCHAR2(100),

catdetalle VARCHAR2(500),

status NUMBER(1) DEFAULT 1,

imagen CLOB

);

ALTER TABLE categoria ADD CONSTRAINT categoria\_pk PRIMARY KEY (catid);

ALTER TABLE categoria ADD (imagen\_temp BLOB);

UPDATE categoria SET imagen\_temp = UTL\_RAW.CAST\_TO\_RAW(imagen);

ALTER TABLE categoria DROP COLUMN imagen;

ALTER TABLE categoria RENAME COLUMN imagen\_temp TO imagen;

-- Insertar categorías en la tabla categoria

INSERT INTO categoria (catid, catname, catdetalle, imagen) VALUES (CAT\_SEQ.NEXTVAL, 'Destacados', 'Productos destacados de la tienda', EMPTY\_BLOB());

INSERT INTO categoria (catid, catname, catdetalle, imagen) VALUES (CAT\_SEQ.NEXTVAL, 'Celulares', 'Teléfonos móviles y accesorios', EMPTY\_BLOB());

INSERT INTO categoria (catid, catname, catdetalle, imagen) VALUES (CAT\_SEQ.NEXTVAL, 'Computación', 'Computadoras, laptops y accesorios de computación', EMPTY\_BLOB());

INSERT INTO categoria (catid, catname, catdetalle, imagen) VALUES (CAT\_SEQ.NEXTVAL, 'Accesorios', 'Accesorios y periféricos para tecnología', EMPTY\_BLOB());

INSERT INTO categoria (catid, catname, catdetalle, imagen) VALUES (CAT\_SEQ.NEXTVAL, 'Smart Home', 'Productos inteligentes para el hogar', EMPTY\_BLOB());

INSERT INTO categoria (catid, catname, catdetalle, imagen) VALUES (CAT\_SEQ.NEXTVAL, 'Gaming', 'Consolas, juegos y accesorios para gamers', EMPTY\_BLOB());

INSERT INTO categoria (catid, catname, catdetalle, imagen) VALUES (CAT\_SEQ.NEXTVAL, 'Audio y Video', 'Parlantes, auriculares, y dispositivos de video', EMPTY\_BLOB());

INSERT INTO categoria (catid, catname, catdetalle, imagen) VALUES (CAT\_SEQ.NEXTVAL, 'Wearables', 'Relojes inteligentes y otros dispositivos vestibles', EMPTY\_BLOB());

INSERT INTO categoria (catid, catname, catdetalle, imagen) VALUES (CAT\_SEQ.NEXTVAL, 'Cámaras y Fotografía', 'Cámaras digitales y equipos fotográficos', EMPTY\_BLOB());

INSERT INTO categoria (catid, catname, catdetalle, imagen) VALUES (CAT\_SEQ.NEXTVAL, 'Drones y Robots', 'Drones y dispositivos robóticos', EMPTY\_BLOB());

-- Tabla producto (necesita categoría)

CREATE TABLE producto (

productid INTEGER DEFAULT PROD\_SEQ.NEXTVAL,

productname VARCHAR2(500),

marca VARCHAR2(100),

descripcion CLOB,

precio NUMBER(10,2),

categoria\_catid INTEGER NOT NULL,

imagen BLOB NULL

);

ALTER TABLE producto ADD CONSTRAINT producto\_pk PRIMARY KEY (productid);

ALTER TABLE producto ADD CONSTRAINT producto\_categoria\_fk FOREIGN KEY (categoria\_catid) REFERENCES categoria (catid);

-- Agregar las columnas inventario, inventario\_disponible y reserva a la tabla producto

ALTER TABLE producto ADD (inventario INTEGER NULL);

ALTER TABLE producto ADD (inventario\_disponible INTEGER NULL);

ALTER TABLE producto ADD (reserva INTEGER NULL);

-- Tabla detalle\_carrito (conectada directamente con usuario y producto)

CREATE TABLE detalle\_carrito (

id\_detalle\_carrito INTEGER DEFAULT DETAIL\_CAR\_SEQ.NEXTVAL,

usuario\_id\_user INTEGER NOT NULL, -- Relación directa con usuario

productid INTEGER NOT NULL, -- Relación con producto

cantidad INTEGER,

costo\_unitario NUMBER(10,2),

costo\_total NUMBER(10,2)

);

ALTER TABLE detalle\_carrito ADD CONSTRAINT detalle\_carrito\_pk PRIMARY KEY (id\_detalle\_carrito);

ALTER TABLE detalle\_carrito ADD CONSTRAINT detalle\_carrito\_usuario\_fk FOREIGN KEY (usuario\_id\_user) REFERENCES usuario (id\_user);

ALTER TABLE detalle\_carrito ADD CONSTRAINT detalle\_carrito\_producto\_fk FOREIGN KEY (productid) REFERENCES producto (productid);

-- Creación de la tabla `pedido` consolidada

CREATE TABLE pedido (

pedidoid INTEGER DEFAULT PED\_SEQ.NEXTVAL,

usuarios\_userid INTEGER NOT NULL,

comuna VARCHAR2(100),

direccion VARCHAR2(100),

detalle VARCHAR2(200),

precio NUMBER(10,2), -- Campo para almacenar el precio del producto

cantidad INTEGER, -- Campo para la cantidad

estado VARCHAR2(100) -- Campo para el estado del pedido

);

-- Clave primaria para `pedido`

ALTER TABLE pedido ADD CONSTRAINT pedido\_pk PRIMARY KEY (pedidoid);

-- Claves foráneas para referenciar a `usuario` y `producto`

ALTER TABLE pedido ADD CONSTRAINT pedido\_usuarios\_fk FOREIGN KEY (usuarios\_userid) REFERENCES usuario (id\_user);

-- Tabla imagen\_producto (necesita producto)

CREATE TABLE imagen\_producto (

imagenid INTEGER DEFAULT IMG\_SEQ.NEXTVAL,

imagen CLOB,

producto\_productid INTEGER NOT NULL,

esprimario NUMBER(1)

);

ALTER TABLE imagen\_producto ADD CONSTRAINT imagen\_producto\_pk PRIMARY KEY (imagenid);

ALTER TABLE imagen\_producto ADD CONSTRAINT imagen\_producto\_producto\_fk FOREIGN KEY (producto\_productid) REFERENCES producto (productid);

-- Tabla movimiento\_kar

CREATE TABLE movimiento\_kar (

movid INTEGER DEFAULT MOV\_SEQ.NEXTVAL,

movname VARCHAR2(100)

);

ALTER TABLE movimiento\_kar ADD CONSTRAINT movimiento\_kar\_pk PRIMARY KEY (movid);

-- Tabla kardex (necesita movimiento\_kar)

CREATE TABLE kardex (

kardexid INTEGER DEFAULT KARDEX\_SEQ.NEXTVAL,

fecha DATE,

hora VARCHAR2(8),

entrada INTEGER,

salida INTEGER,

saldo INTEGER,

preciounitario NUMBER(10,2),

precioactualizado NUMBER(10,2),

movimiento\_kar\_movid INTEGER NOT NULL,

producto\_productid INTEGER -- Agregar la relación con producto

);

ALTER TABLE kardex ADD CONSTRAINT kardex\_pk PRIMARY KEY (kardexid);

ALTER TABLE kardex ADD CONSTRAINT kardex\_movimiento\_kar\_fk FOREIGN KEY (movimiento\_kar\_movid) REFERENCES movimiento\_kar (movid);

ALTER TABLE kardex ADD CONSTRAINT kardex\_producto\_fk FOREIGN KEY (producto\_productid) REFERENCES producto (productid);

-- Tabla direccion (necesita usuario)

CREATE TABLE direccion (

dirid INTEGER DEFAULT DIR\_SEQ.NEXTVAL,

comuna VARCHAR2(100),

direccion VARCHAR2(100),

detalle VARCHAR2(200),

dirprincipal NUMBER(1),

usuario\_id\_user INTEGER NOT NULL

);

ALTER TABLE direccion ADD CONSTRAINT direccion\_pk PRIMARY KEY (dirid);

ALTER TABLE direccion ADD CONSTRAINT direccion\_usuario\_fk FOREIGN KEY (usuario\_id\_user) REFERENCES usuario (id\_user);

-- Tabla valoracion (necesita usuario y producto)

CREATE TABLE valoracion (

valid INTEGER DEFAULT VAL\_SEQ.NEXTVAL,

valcomentario CLOB,

valpuntuacion INTEGER,

usuarios\_userid INTEGER NOT NULL,

producto\_productid INTEGER NOT NULL

);

ALTER TABLE valoracion ADD CONSTRAINT valoracion\_pk PRIMARY KEY (valid);

ALTER TABLE valoracion ADD CONSTRAINT valoracion\_usuarios\_fk FOREIGN KEY (usuarios\_userid) REFERENCES usuario (id\_user);

ALTER TABLE valoracion ADD CONSTRAINT valoracion\_producto\_fk FOREIGN KEY (producto\_productid) REFERENCES producto (productid);

-- Tabla etiqueta

CREATE TABLE etiqueta (

etiquetaId INTEGER DEFAULT ETIQUETA\_SEQ.NEXTVAL,

nombreEtiqueta VARCHAR2(100), -- Texto medianamente largo

detalleEtiqueta CLOB, -- Texto largo para descripciones extensas

CONSTRAINT pk\_etiqueta PRIMARY KEY (etiquetaId)

);

-- Insertar etiquetas en la tabla etiqueta

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'Procesador', 'El procesador es el cerebro del computador. Es responsable de ejecutar las instrucciones y realizar cálculos necesarios para que el sistema funcione. Es esencial para la velocidad y el rendimiento.');

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'Placa Madre', 'La placa madre es el componente principal donde se conectan todos los demás. Permite que el procesador, memoria, almacenamiento y otros dispositivos trabajen juntos.');

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'Ventilador CPU', 'El ventilador del CPU mantiene al procesador frío mientras trabaja.');

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'Memoria RAM', 'La memoria RAM almacena temporalmente los datos que el procesador necesita mientras ejecuta programas. Cuanta más RAM, mejor será el rendimiento en tareas multitarea.');

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'Tarjeta Gráfica', 'La tarjeta gráfica maneja todo lo relacionado con el renderizado de imágenes y videos. Es crucial para juegos, diseño gráfico y otras tareas intensivas en gráficos.');

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'Disco Duro SSD', 'Un disco duro SSD (unidad de estado sólido) es más rápido y fiable que un disco duro tradicional. Acelera el arranque del sistema y la carga de programas.');

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'Fuente de Poder', 'La fuente de poder proporciona la electricidad necesaria para que todos los componentes de la computadora funcionen correctamente.');

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'Caja PC', 'La caja de la PC es la estructura que alberga todos los componentes internos del equipo, proporcionando ventilación y seguridad.');

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'Teclado', 'El teclado es uno de los dispositivos de entrada más comunes que permite al usuario interactuar con la computadora.');

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'Mouse', 'El mouse es un dispositivo apuntador que permite controlar el cursor en la pantalla de la computadora.');

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL,'Socket AM5', 'Introducción:

El socket AM5 fue lanzado en 2022 junto con los procesadores AMD Ryzen 7000 Series, basados en la arquitectura Zen 4.

Es el sucesor del socket AM4, que tuvo una larga vida útil y soportó múltiples generaciones de procesadores.

Tipo de socket:

LGA (Land Grid Array): A diferencia del AM4, que era PGA (Pin Grid Array), el AM5 utiliza LGA, lo que significa que los pines están en el socket de la placa madre, no en el procesador. El formato específico es LGA1718, con 1718 pines de contacto.

Compatibilidad de memoria:

DDR5: Solo es compatible con memorias DDR5, lo que marca un avance significativo en velocidad y ancho de banda en comparación con el soporte DDR4 del socket AM4.

Compatibilidad de PCIe:

PCIe 5.0: Ofrece soporte para dispositivos PCIe de quinta generación, lo que permite un aumento en la velocidad de transferencia para almacenamiento y tarjetas gráficas.

Compatibilidad térmica y de refrigeración:

Aunque es un nuevo socket, mantiene la compatibilidad con los sistemas de refrigeración diseñados para el AM4. Esto facilita las actualizaciones sin necesidad de cambiar el cooler en muchos casos.

Plataformas asociadas:

Funciona con las placas madre basadas en los chipsets AMD 600 Series, como B650, B650E, X670 y X670E.

Ventajas principales:

Soporte para tecnologías más nuevas como DDR5, PCIe 5.0 y mayor eficiencia energética.

Diseño mejorado con mayor cantidad de pines, permitiendo una mayor capacidad de alimentación para los procesadores de alto rendimiento.

Longevidad esperada:

AMD ha anunciado que planea darle al AM5 una vida útil larga, similar a lo que hizo con el AM4, lo que es excelente para quienes buscan un camino de actualización a futuro.');

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'CPU AM5', 'Introducción:

La plataforma AM5 está diseñada específicamente para los procesadores de la serie AMD Ryzen 7000 y futuros modelos, basados en la arquitectura Zen 4 y posteriores. Este socket es parte del compromiso de AMD para ofrecer mejoras significativas en rendimiento, eficiencia y compatibilidad.

Compatibilidad:

La CPU AM5 es compatible exclusivamente con placas base que soportan el socket AM5. Este estándar permite utilizar procesadores de alto rendimiento como los Ryzen 7000, así como futuros chips de la familia Ryzen.

Tecnologías soportadas:

- DDR5: La nueva plataforma permite el uso de memorias DDR5, lo que mejora significativamente el rendimiento en comparación con las generaciones anteriores.

- PCIe 5.0: Ofrece soporte para tarjetas gráficas y almacenamiento más rápidos, lo que aumenta las capacidades de expansión y rendimiento general del sistema.

Rendimiento:

La CPU AM5 ofrece una mejora de la eficiencia energética y una mayor capacidad de procesamiento para tareas exigentes como juegos, edición de video y trabajo con aplicaciones profesionales.

Ventajas principales:

- Soporte para las últimas tecnologías, como DDR5 y PCIe 5.0.

- Mayor rendimiento en aplicaciones multi-threading y tareas exigentes.

- Compatible con placas base de chipsets AMD 600 Series (B650, B650E, X670, X670E).

Longevidad esperada:

Se espera que la plataforma AM5 ofrezca un ciclo de vida prolongado, permitiendo que los usuarios actualicen sus procesadores sin necesidad de cambiar la placa base por un largo tiempo.');

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'Placa AM5', 'Introducción:

Las placas base AM5 están diseñadas para soportar los procesadores de la serie AMD Ryzen 7000 y otros futuros procesadores basados en la arquitectura Zen 4 y posteriores. Estas placas son esenciales para aprovechar al máximo las nuevas características y tecnologías de la plataforma AM5.

Compatibilidad de procesadores:

Las placas AM5 son compatibles únicamente con los procesadores Ryzen 7000 y futuros chips que utilicen el socket AM5, garantizando un alto rendimiento y la posibilidad de actualización en el futuro.

Chipsets compatibles:

Las placas AM5 son compatibles con los chipsets AMD 600 Series, que incluyen opciones como el B650, B650E, X670 y X670E. Estos chipsets permiten configuraciones avanzadas de overclocking, soporte para múltiples tarjetas gráficas y almacenamiento ultrarrápido.

Tecnologías soportadas:

- DDR5: Soporta memoria DDR5 para un aumento de la velocidad y el ancho de banda en comparación con DDR4.

- PCIe 5.0: Proporciona soporte para la última generación de dispositivos de expansión, lo que mejora las velocidades de transferencia de datos para tarjetas gráficas y almacenamiento.

Ventajas principales:

- Soporte para la plataforma AM5, permitiendo el uso de procesadores Ryzen 7000 y más.

- Amplias opciones de conectividad, incluyendo USB 4.0, PCIe 5.0 y soporte para RAID.

- Capacidades avanzadas de overclocking, aumentando aún más el rendimiento del sistema.

Longevidad esperada:

Las placas base AM5 están diseñadas para soportar las futuras generaciones de procesadores y tecnologías, lo que proporciona a los usuarios una plataforma sólida y actualizable para el futuro.');

-- Insertar Socket LGA1700

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'Socket LGA1700', 'Introducción:

El socket LGA1700 es un zócalo utilizado para los procesadores Intel de 12ª y 13ª generación, como los de la serie Alder Lake y Raptor Lake. Fue lanzado en 2021 y es un avance respecto al LGA1200, ofreciendo más pines para un mejor rendimiento y eficiencia.

Tipo de socket:

LGA (Land Grid Array): En este diseño, los pines están en el zócalo de la placa madre, no en el procesador. El socket LGA1700 tiene 1700 pines de contacto, lo que permite una mayor capacidad de comunicación entre el procesador y otros componentes.

Compatibilidad de memoria:

DDR4 y DDR5: El socket LGA1700 es compatible tanto con memoria DDR4 como DDR5, lo que permite una mayor flexibilidad en cuanto a la velocidad de la memoria y el rendimiento general del sistema.

Compatibilidad de PCIe:

PCIe 5.0: Ofrece soporte para dispositivos PCIe de quinta generación, lo que mejora las velocidades de transferencia, permitiendo el uso de tarjetas gráficas y almacenamiento de alto rendimiento.

Compatibilidad térmica y de refrigeración:

El LGA1700 es un socket de mayor tamaño que el LGA1200, por lo que los sistemas de refrigeración deben ser compatibles con este nuevo formato para garantizar un rendimiento térmico adecuado.

Plataformas asociadas:

Funciona con placas base que utilizan chipsets como el Z690, B660, H670 y Z790, entre otros.

Ventajas principales:

- Mejor rendimiento con memoria DDR5 y PCIe 5.0.

- Mayor capacidad de alimentación para procesadores de alto rendimiento.

Longevidad esperada:

Se espera que el LGA1700 sea una plataforma duradera para futuras generaciones de procesadores Intel, permitiendo una vida útil extendida y más opciones de actualización.');

-- Insertar CPU LGA1700

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'CPU LGA1700', 'Introducción:

La CPU LGA1700 es una serie de procesadores Intel de 12ª y 13ª generación, como los de las familias Alder Lake y Raptor Lake. Estos procesadores están diseñados para ofrecer un alto rendimiento en tareas multitarea y aplicaciones intensivas, como juegos, edición de video y trabajo profesional.

Compatibilidad:

La CPU LGA1700 es exclusiva para placas base con zócalo LGA1700. Esto garantiza una conexión estable y eficiente entre el procesador y el resto de los componentes del sistema.

Tecnologías soportadas:

- DDR4 y DDR5: Los procesadores de la serie LGA1700 son compatibles con memorias DDR4 y DDR5, lo que proporciona mayor ancho de banda y velocidad de transferencia.

- PCIe 5.0: Los procesadores de LGA1700 también ofrecen soporte para la última generación de dispositivos PCIe, mejorando las capacidades de expansión del sistema.

Rendimiento:

Los procesadores LGA1700 ofrecen una arquitectura híbrida, combinando núcleos de alto rendimiento con núcleos de eficiencia para un mejor equilibrio entre rendimiento y consumo de energía.

Ventajas principales:

- Rendimiento superior para tareas exigentes como juegos y edición.

- Soporte para las últimas tecnologías, como DDR5 y PCIe 5.0.

Longevidad esperada:

Intel ha diseñado estos procesadores con la intención de ofrecer una plataforma de alto rendimiento durante varios años, lo que permite actualizaciones futuras sin necesidad de cambiar la placa base.');

-- Insertar Placa LGA1700

INSERT INTO etiqueta (etiquetaId, nombreEtiqueta, detalleEtiqueta)

VALUES (ETIQUETA\_SEQ.NEXTVAL, 'Placa LGA1700', 'Introducción:

Las placas base LGA1700 están diseñadas para soportar los procesadores Intel de 12ª y 13ª generación, como los de las series Alder Lake y Raptor Lake. Estas placas base son fundamentales para aprovechar el rendimiento de los nuevos procesadores y las tecnologías avanzadas como DDR5 y PCIe 5.0.

Compatibilidad de procesadores:

Las placas base LGA1700 son compatibles solo con los procesadores Intel que utilizan este socket, como los de la familia Alder Lake y Raptor Lake.

Chipsets compatibles:

Las placas base LGA1700 utilizan chipsets como el Z690, B660, Z790 y H670, que ofrecen soporte para overclocking, almacenamiento ultrarrápido y múltiples configuraciones de tarjetas gráficas.

Tecnologías soportadas:

- DDR4 y DDR5: Soporta ambos tipos de memoria, permitiendo una mayor flexibilidad según el rendimiento que se desee.

- PCIe 5.0: Proporciona soporte para la última generación de tarjetas gráficas, almacenamiento y otros dispositivos, mejorando la velocidad de transferencia de datos.

Ventajas principales:

- Soporte para tecnologías avanzadas como DDR5 y PCIe 5.0.

- Mejores opciones de conectividad y expansión, incluyendo soporte para múltiples dispositivos PCIe.

Longevidad esperada:

Estas placas base están diseñadas para ofrecer soporte a futuras generaciones de procesadores Intel, proporcionando una plataforma de actualización duradera y robusta.');

-- Tabla etiqueta\_producto

CREATE TABLE etiqueta\_producto (

etiquetaProductoId INTEGER DEFAULT ETIQUETA\_PRODUCTO\_SEQ.NEXTVAL,

productid INTEGER, -- Clave foránea a producto

etiquetaId INTEGER, -- Clave foránea a etiqueta

CONSTRAINT pk\_etiqueta\_producto PRIMARY KEY (etiquetaProductoId),

CONSTRAINT fk\_etiqueta\_producto\_producto FOREIGN KEY (productid) REFERENCES producto (productid),

CONSTRAINT fk\_etiqueta\_producto\_etiqueta FOREIGN KEY (etiquetaId) REFERENCES etiqueta (etiquetaId)

);

-- Cambios en pedido

ALTER TABLE pedido

ADD (

orderId VARCHAR2(40), -- Identificador único de la transacción de pago

currency VARCHAR2(10) DEFAULT 'CLP', -- Moneda del pago

urlReturn VARCHAR2(200), -- URL de retorno después del pago

urlNotify VARCHAR2(200) -- URL de notificación para el pago

);

CREATE TABLE PEDIDO\_PRODUCTO (

ID NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

PEDIDO\_ID NUMBER NOT NULL,

PRODUCTO\_ID NUMBER NOT NULL,

CANTIDAD NUMBER NOT NULL,

PRECIO\_UNITARIO DECIMAL(10, 2) NOT NULL,

TOTAL\_PRECIO DECIMAL(10, 2) NOT NULL,

CONSTRAINT FK\_PEDIDO FOREIGN KEY (PEDIDO\_ID) REFERENCES PEDIDO(PEDIDOID),

CONSTRAINT FK\_PRODUCTO FOREIGN KEY (PRODUCTO\_ID) REFERENCES PRODUCTO(PRODUCTID)

);

-- Tabla guia

CREATE TABLE guia (

guiaid INTEGER DEFAULT GUIA\_SEQ.NEXTVAL,

detalle CLOB

);

ALTER TABLE guia ADD CONSTRAINT guia\_pk PRIMARY KEY (guiaid);

ALTER TABLE PEDIDO MODIFY DETALLE VARCHAR2(255);

ALTER TABLE PRODUCTO

ADD status NUMBER(1) DEFAULT 1;