-- 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;

-- 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),

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(70),

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);

-- 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,

producto\_productid INTEGER NOT NULL,

comuna VARCHAR2(100),

direccion VARCHAR2(100),

detalle VARCHAR2(100),

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);

ALTER TABLE pedido ADD CONSTRAINT pedido\_producto\_fk FOREIGN KEY (producto\_productid) REFERENCES producto (productid);  
  
-- Tabla detallepedido (necesita pedido)

CREATE TABLE detallepedido (

detpedid INTEGER DEFAULT DETAIL\_PED\_SEQ.NEXTVAL,

preciounitario NUMBER(10,2),

cantidad INTEGER,

pedido\_pedidoid INTEGER NOT NULL

);

ALTER TABLE detallepedido ADD CONSTRAINT detallepedido\_pk PRIMARY KEY (detpedid);

ALTER TABLE detallepedido ADD CONSTRAINT detallepedido\_pedido\_fk FOREIGN KEY (pedido\_pedidoid) REFERENCES pedido (pedidoid);

-- 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(100),

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)

);

-- 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)

);