

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS  
SISTEMAS DE BASES DE DATOS 1  
ING. ÁLVARO GIOVANNI LONGO MORALES  
TUTOR: JOSÉ VALERIO CHOC MIJANGOS

## Práctica 2

Grupo 9

Nombre	Carnet
Daniel Eduardo Mellado Ayala	201020232
Piter Angel Esaú Valiente de León	201902301
Eduardo Josué González Cifuentes	201900647

## ● Análisis

### Entidades:

#### 1. Producto:

- Atributos:

- id\_producto(pk)
- Nombre
- Precio

- Relaciones:

- Un producto pertenece a una única categoría, pero una categoría puede tener muchos productos.

#### 2. Cliente:

- Atributos:

- id\_cliente(pk)
- Nombre
- Apellido
- Dirección
- Teléfono
- Tarjeta
- Edad
- Salario
- Género

- Relaciones:

- Varios clientes pueden pertenecer a un país.

#### 3. Vendedor:

- Atributos:

- id\_vendedor(pk)
- Nombre

- Relaciones:

- Varios vendedores pueden pertenecer a un país.

#### 4. Orden de Venta:

- Atributos:

- id\_orden(pk)
- Linea\_orden
- Fecha\_orden
- Cantidad

- Relaciones:

- Un cliente puede realizar varias órdenes de venta, pero cada orden de venta está asociada a un único cliente.
- Un vendedor puede realizar varias órdenes de venta, pero cada orden de venta está asociada a un único vendedor.
- Una orden de venta puede contener múltiples productos, y un producto puede ser vendido en múltiples órdenes de venta.

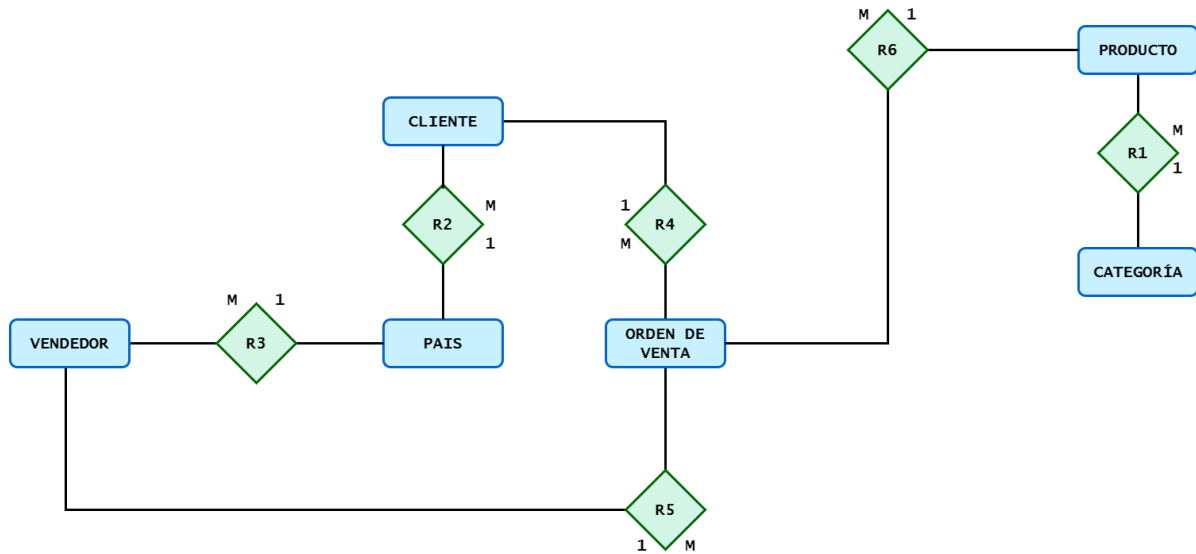
## 5. Categoría

- Atributos:
  - id\_categoria(pk)
  - Nombre

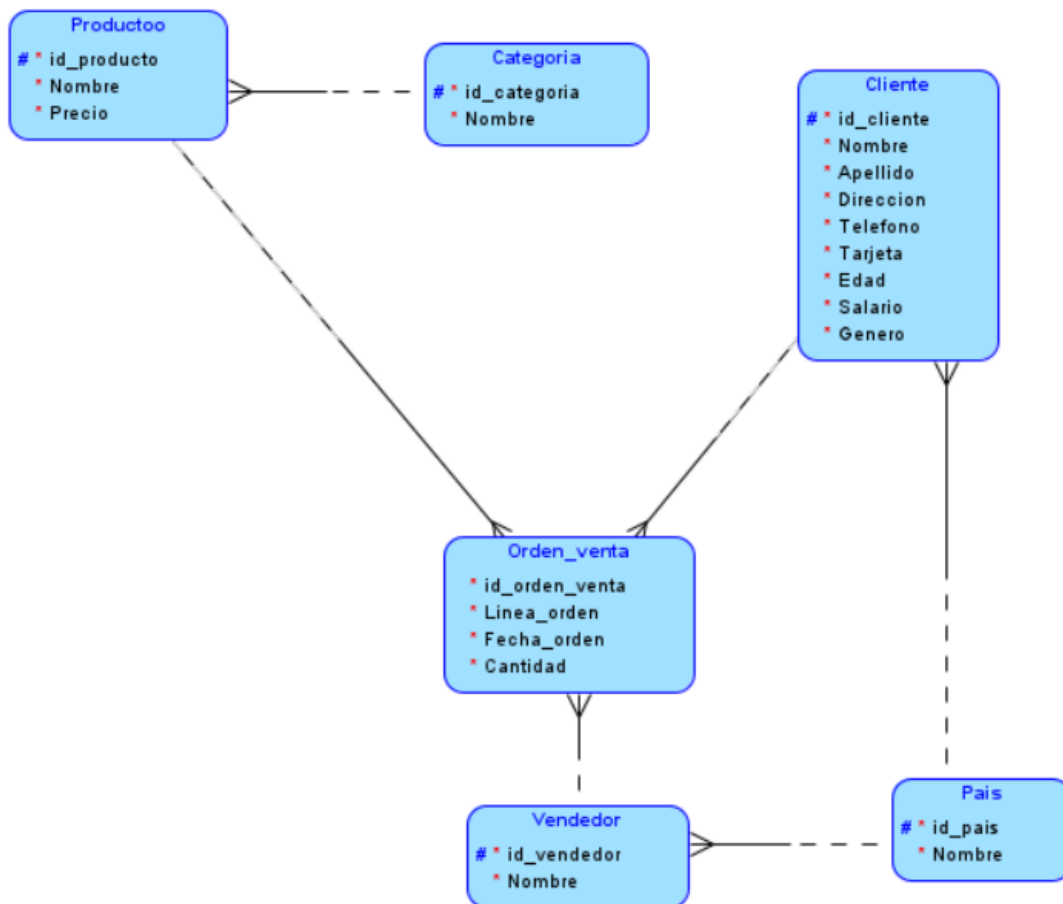
## 6. País:

- Atributos:
  - id\_pais(pk)
  - Nombre

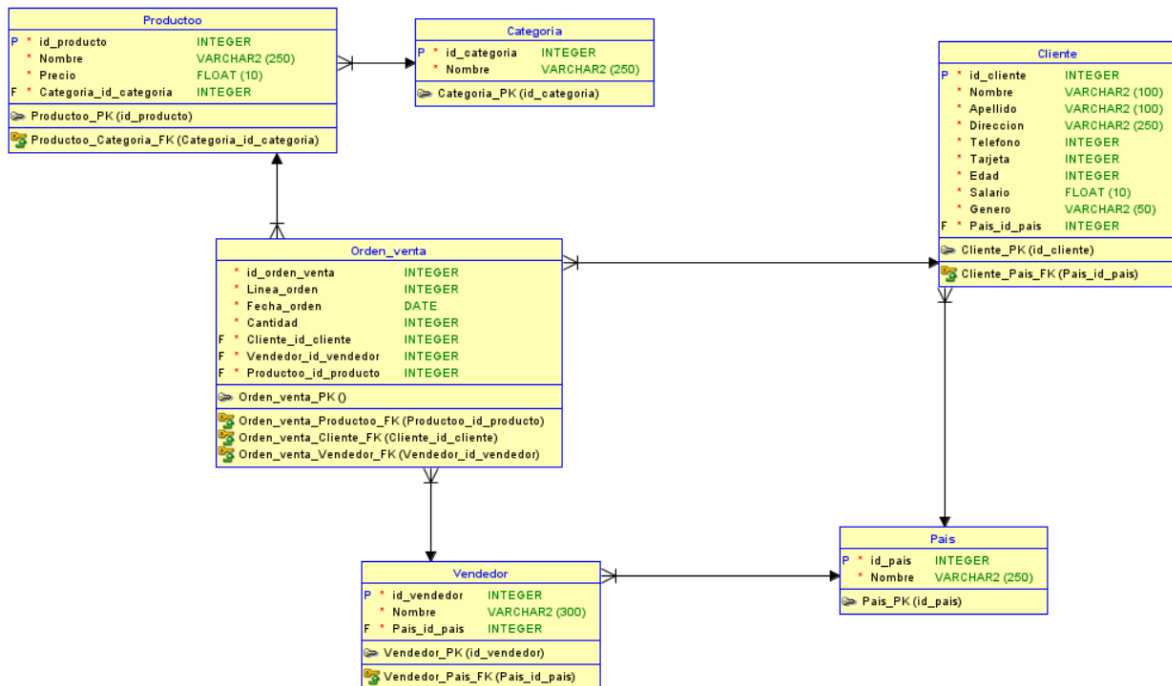
## • Modelo Conceptual



## • Modelo Lógico



## ● Modelo Relacional



## • Scripts

```
-- Generado por Oracle SQL Developer Data Modeler 23.1.0.087.0806
-- en:      2024-04-10 17:20:56 CST
-- sitio:   Oracle Database 21c
-- tipo:    Oracle Database 21c
```

```
-- predefined type, no DDL - MDSYS.SDO_GEOMETRY
```

```
-- predefined type, no DDL - XMLTYPE
```

```
CREATE TABLE categoria (
  id_categoria INTEGER NOT NULL,
  nombre       VARCHAR2(250) NOT NULL
);
```

```
ALTER TABLE categoria ADD CONSTRAINT categoria_pk PRIMARY KEY (
id_categoria );
```

```
CREATE TABLE cliente (
  id_cliente  INTEGER NOT NULL,
  nombre      VARCHAR2(100) NOT NULL,
  apellido    VARCHAR2(100) NOT NULL,
  direccion   VARCHAR2(250) NOT NULL,
  telefono    INTEGER NOT NULL,
  tarjeta     INTEGER NOT NULL,
  edad        INTEGER NOT NULL,
  salario     FLOAT(10) NOT NULL,
  genero      VARCHAR2(50) NOT NULL,
  pais_id_pais INTEGER NOT NULL
);
```

```
ALTER TABLE cliente ADD CONSTRAINT cliente_pk PRIMARY KEY (
id_cliente );
```

```
CREATE TABLE orden_venta (
  id_orden_venta  INTEGER NOT NULL,
  linea_orden     INTEGER NOT NULL,
  fecha_orden     DATE NOT NULL,
  cantidad        INTEGER NOT NULL,
  cliente_id_cliente INTEGER NOT NULL,
  vendedor_id_vendedor INTEGER NOT NULL,
```

```
productoo_id_producto INTEGER NOT NULL  
);
```

```
CREATE TABLE pais (  
  id_pais INTEGER NOT NULL,  
  nombre VARCHAR2(250) NOT NULL  
);
```

```
ALTER TABLE pais ADD CONSTRAINT pais_pk PRIMARY KEY ( id_pais );
```

```
CREATE TABLE productoo (  
  id_producto      INTEGER NOT NULL,  
  nombre           VARCHAR2(250) NOT NULL,  
  precio           FLOAT(10) NOT NULL,  
  categoria_id_categoria INTEGER NOT NULL  
);
```

```
ALTER TABLE productoo ADD CONSTRAINT productoo_pk PRIMARY KEY  
( id_producto );
```

```
CREATE TABLE vendedor (  
  id_vendedor INTEGER NOT NULL,  
  nombre      VARCHAR2(300) NOT NULL,  
  pais_id_pais INTEGER NOT NULL  
);
```

```
ALTER TABLE vendedor ADD CONSTRAINT vendedor_pk PRIMARY KEY (  
id_vendedor );
```

```
ALTER TABLE cliente  
  ADD CONSTRAINT cliente_pais_fk FOREIGN KEY ( pais_id_pais )  
  REFERENCES pais ( id_pais );
```

```
ALTER TABLE orden_venta  
  ADD CONSTRAINT orden_venta_cliente_fk FOREIGN KEY (  
cliente_id_cliente )  
  REFERENCES cliente ( id_cliente );
```

```
ALTER TABLE orden_venta  
  ADD CONSTRAINT orden_venta_productoo_fk FOREIGN KEY (  
productoo_id_producto )  
  REFERENCES productoo ( id_producto );
```

```
ALTER TABLE orden_venta
```

```

ADD CONSTRAINT orden_venta_vendedor_fk FOREIGN KEY (
vendedor_id_vendedor )
REFERENCES vendedor ( id_vendedor );

```

```

ALTER TABLE productoo
ADD CONSTRAINT productoo_categoria_fk FOREIGN KEY (
categoria_id_categoria )
REFERENCES categoria ( id_categoria );

```

```

ALTER TABLE vendedor
ADD CONSTRAINT vendedor_pais_fk FOREIGN KEY ( pais_id_pais )
REFERENCES pais ( id_pais );

```

-- Informe de Resumen de Oracle SQL Developer Data Modeler:

```

--
-- CREATE TABLE                6
-- CREATE INDEX                 0
-- ALTER TABLE                11
-- CREATE VIEW                  0
-- ALTER VIEW                   0
-- CREATE PACKAGE                0
-- CREATE PACKAGE BODY           0
-- CREATE PROCEDURE              0
-- CREATE FUNCTION              0
-- CREATE TRIGGER                0
-- ALTER TRIGGER                 0
-- CREATE COLLECTION TYPE        0
-- CREATE STRUCTURED TYPE        0
-- CREATE STRUCTURED TYPE BODY   0
-- CREATE CLUSTER                0
-- CREATE CONTEXT                0
-- CREATE DATABASE               0
-- CREATE DIMENSION              0
-- CREATE DIRECTORY              0
-- CREATE DISK GROUP             0
-- CREATE ROLE                   0
-- CREATE ROLLBACK SEGMENT       0
-- CREATE SEQUENCE               0
-- CREATE MATERIALIZED VIEW      0
-- CREATE MATERIALIZED VIEW LOG  0
-- CREATE SYNONYM                0
-- CREATE TABLESPACE            0
-- CREATE USER                   0

```



```
--
-- DROP TABLESPACE          0
-- DROP DATABASE             0
--
-- REDACTION POLICY          0
--
-- ORDS DROP SCHEMA          0
-- ORDS ENABLE SCHEMA        0
-- ORDS ENABLE OBJECT        0
--
-- ERRORS                     0
-- WARNINGS                   0
```

## -- CONSULTAS

--1 BIEN

```
SELECT c.id_cliente, c.nombre, c.apellido, p.nombre AS país,
SUM(o.cantidad * pr.precio) AS monto_total
FROM cliente c
JOIN orden_venta o ON c.id_cliente = o.cliente_id_cliente
JOIN pais p ON c.pais_id_pais = p.id_pais
JOIN productoo pr ON o.productoo_id_producto = pr.id_producto
GROUP BY c.id_cliente, c.nombre, c.apellido, p.nombre
ORDER BY monto_total DESC
FETCH FIRST 1 ROW ONLY;
```

-- 2 BUENO

```
(
  SELECT p.id_producto, p.nombre, c.nombre AS Categoria, p.precio,
SUM(o.cantidad) AS cantidad, SUM(o.cantidad * p.precio) AS monto
  FROM orden_venta o
  JOIN productoo p ON p.id_producto = o.productoo_id_producto
  JOIN categoria c ON c.id_categoria = p.categoria_id_categoria
  GROUP BY p.id_producto, p.nombre, c.nombre, p.precio
  ORDER BY cantidad, monto ASC
  FETCH NEXT 1 ROWS ONLY
)
UNION ALL
(
  SELECT p.id_producto, p.nombre, c.nombre AS Categoria, p.precio,
SUM(o.cantidad) AS cantidad, SUM(o.cantidad * p.precio) AS monto
  FROM orden_venta o
```

```

JOIN productoo p ON p.id_producto = o.productoo_id_producto
JOIN categoria c ON c.id_categoria = p.categoria_id_categoria
GROUP BY p.id_producto, p.nombre, c.nombre, p.precio
ORDER BY cantidad DESC
FETCH NEXT 1 ROWS ONLY
);

```

-- 3 BUENO

-- Crear la vista Ventas\_Personas  
CREATE VIEW Ventas\_Personas AS

```

SELECT
    vendedor.id_vendedor,
    vendedor.nombre,
    pais.nombre AS pais,
    SUM(productoo.precio * orden_venta.cantidad) AS monto
FROM
    orden_venta
JOIN
    vendedor ON vendedor.id_vendedor =
orden_venta.vendedor_id_vendedor
JOIN
    pais ON pais.id_pais = vendedor.pais_id_pais
JOIN
    productoo ON productoo.id_producto =
orden_venta.productoo_id_producto
GROUP BY
    vendedor.id_vendedor, vendedor.nombre, pais.nombre;

```

-- Consulta para obtener los vendedores con la venta mínima y máxima

```

SELECT * FROM Ventas_Personas
WHERE monto = (SELECT MIN(monto) FROM Ventas_Personas)
UNION ALL
SELECT * FROM Ventas_Personas
WHERE monto = (SELECT MAX(monto) FROM Ventas_Personas);

```

--4 BUENO

```

WITH VentasPorPais AS (
    SELECT p.id_pais,
           p.nombre AS nombre_pais,
           SUM(pr.precio * ov.cantidad) AS monto
    FROM orden_venta ov
    INNER JOIN vendedor v ON ov.vendedor_id_vendedor = v.id_vendedor
    INNER JOIN pais p ON v.pais_id_pais = p.id_pais

```

```

        INNER JOIN productoo pr ON ov.productoo_id_producto =
pr.id_producto
        GROUP BY p.id_pais, p.nombre
    )
    SELECT nombre_pais, monto
FROM (
    SELECT nombre_pais, monto, ROW_NUMBER() OVER (ORDER BY
monto ASC) AS ranking_asc,
        ROW_NUMBER() OVER (ORDER BY monto DESC) AS ranking_desc
    FROM VentasPorPais
) ranked
WHERE ranking_asc = 1 OR ranking_desc = 1;

```

--5 BIEN

```

SELECT p.id_pais, p.nombre AS país, SUM(o.cantidad * pr.precio) AS
monto_total
FROM pais p
JOIN cliente c ON p.id_pais = c.pais_id_pais
JOIN orden_venta o ON c.id_cliente = o.cliente_id_cliente
JOIN productoo pr ON o.productoo_id_producto = pr.id_producto
GROUP BY p.id_pais, p.nombre
ORDER BY SUM(o.cantidad * pr.precio) DESC
FETCH FIRST 5 ROWS ONLY;

```

--6 BUENO

```

WITH CantidadPorCategoria AS (
    SELECT categoria.id_categoria,
        categoria.nombre,
        SUM(orden_venta.cantidad) AS cantidad
    FROM orden_venta
    JOIN productoo ON productoo.id_producto =
orden_venta.productoo_id_producto
    JOIN categoria ON categoria.id_categoria =
productoo.categoria_id_categoria
    GROUP BY categoria.id_categoria, categoria.nombre
)
SELECT id_categoria, nombre, cantidad
FROM (
    SELECT id_categoria, nombre, cantidad
    FROM CantidadPorCategoria
    ORDER BY cantidad DESC
    FETCH NEXT 1 ROW ONLY
)
UNION ALL

```

```

SELECT id_categoria, nombre, cantidad
FROM (
    SELECT id_categoria, nombre, cantidad
    FROM CantidadPorCategoria
    ORDER BY cantidad ASC
    FETCH NEXT 1 ROW ONLY
);

```

```

--7 BUENO
CREATE VIEW orden AS
SELECT pais.nombre AS pais, categoria.nombre AS categoria,
SUM(orden_venta.cantidad) AS cantidad
FROM orden_venta
JOIN cliente ON cliente.id_cliente = orden_venta.cliente_id_cliente
JOIN productoo ON productoo.id_producto =
orden_venta.productoo_id_producto
JOIN categoria ON categoria.id_categoria =
productoo.categoria_id_categoria
JOIN pais ON pais.id_pais = cliente.pais_id_pais
GROUP BY pais.nombre, categoria.nombre;

WITH VentasConRango AS (
    SELECT pais, categoria, cantidad,
    ROW_NUMBER() OVER (PARTITION BY pais ORDER BY cantidad DESC)
AS rango
    FROM orden
)
SELECT pais, categoria, cantidad
FROM VentasConRango
WHERE rango = 1
ORDER BY pais ASC;

```

```

-- 8 BUENO
SELECT TO_CHAR(orden_venta.fecha_orden, 'MM') AS Mes,

    SUM(orden_venta.cantidad) AS Cantidad,
    SUM(orden_venta.cantidad * productoo.precio) AS Monto
FROM orden_venta

```

```
JOIN vendedor ON vendedor.id_vendedor =  
orden_venta.vendedor_id_vendedor  
JOIN productoo ON productoo.id_producto =  
orden_venta.productoo_id_producto  
JOIN pais ON pais.id_pais = vendedor.pais_id_pais  
WHERE pais.id_pais = 10 -- Inglaterra  
GROUP BY Mes, pais.nombre  
ORDER BY Mes;
```

-- 9 BUENO

```
WITH VentasPorMes AS (  
    SELECT TO_CHAR(orden_venta.fecha_orden, 'MM') AS Mes,  
           SUM(orden_venta.cantidad) AS cantidad,  
           SUM(orden_venta.cantidad * productoo.precio) AS monto  
    FROM orden_venta  
    JOIN vendedor ON vendedor.id_vendedor =  
orden_venta.vendedor_id_vendedor  
    JOIN productoo ON productoo.id_producto =  
orden_venta.productoo_id_producto  
    GROUP BY TO_CHAR(orden_venta.fecha_orden, 'MM')  
)
```

```
(SELECT Mes, cantidad, monto  
FROM VentasPorMes  
ORDER BY cantidad DESC  
FETCH FIRST ROW ONLY)
```

UNION ALL

```
(SELECT Mes, cantidad, monto  
FROM VentasPorMes  
ORDER BY cantidad  
FETCH FIRST ROW ONLY);
```

--10 XD

```
SELECT productoo.id_producto, productoo.nombre,  
SUM(orden_venta.cantidad) AS cantidad, SUM(orden_venta.cantidad *  
productoo.precio) AS monto  
FROM orden_venta  
JOIN productoo ON productoo.id_producto =  
orden_venta.productoo_id_producto  
JOIN categoria ON categoria.id_categoria =  
productoo.categoria_id_categoria  
WHERE categoria.id_categoria = 1 -- es el id de la categoría acción  
GROUP BY productoo.id_producto, productoo.nombre  
ORDER BY productoo.nombre ASC;
```