

Base de datos Empresa

Diagrama E-R

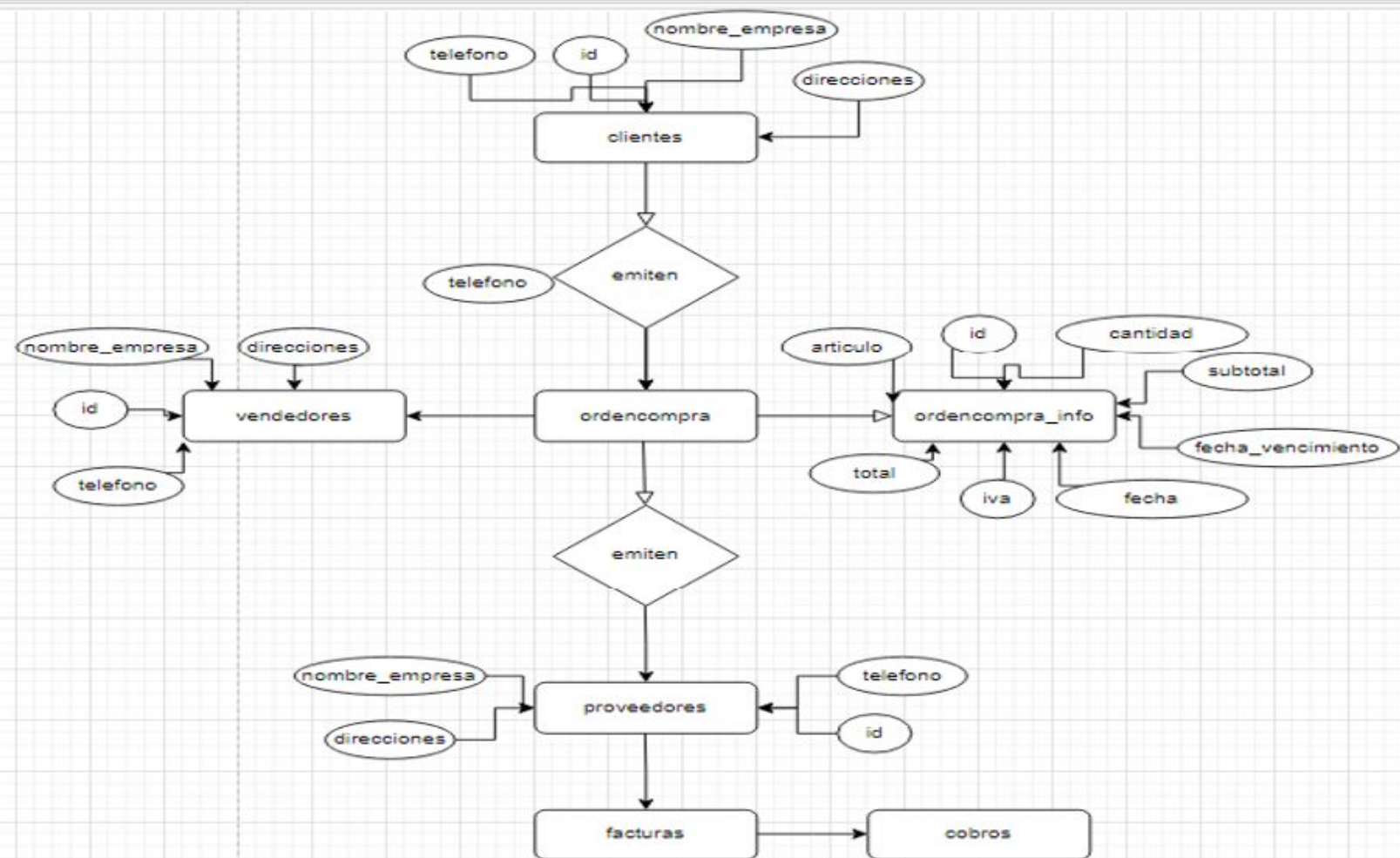
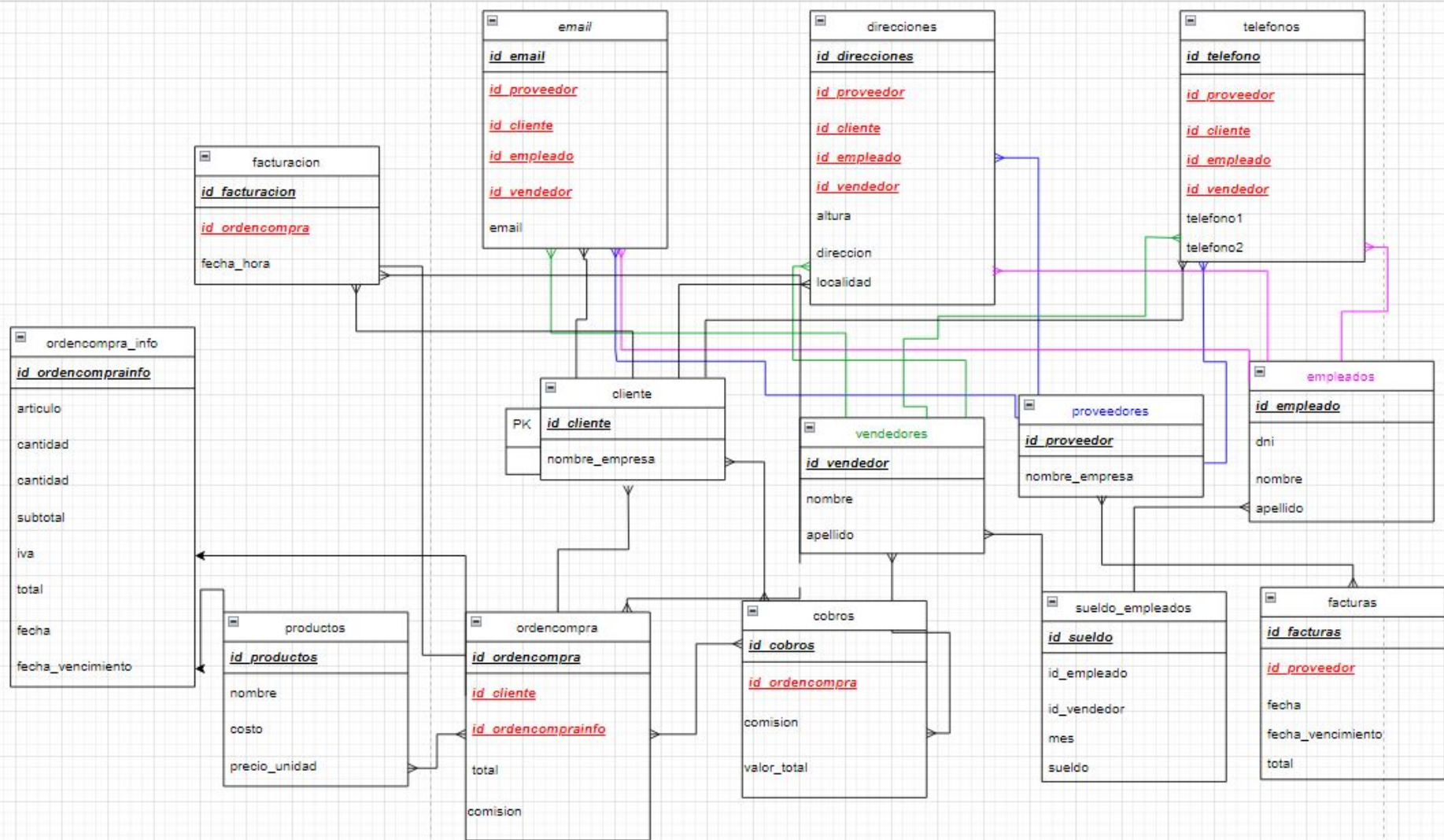


Diagrama Lógico



VIEWS

vs_informacionclientes, el objetivo de esta view es unificar en una tabla el cliente con su respectivo email.

```
CREATE OR REPLACE VIEW vs_informacionclientes AS  
(SELECT id_cliente,nonbre_empresa  
FROM clientes AS C  
JOIN email AS E  ON (E.id_cliente = C.id_cliente))
```

Functions

la funcion calcular_total, espera como parametro el id del cliente y suma todas las Ordenes de compra que cliente haya realizado

```
USE empresa
DELIMITER $$
CREATE FUNCTION calcular_total (cliente INT)
RETURNS DECIMAL(11,2)
READS SQL DATA
BEGIN
    DECLARE resultado DECIMAL(11,2);
    DECLARE id_cliente INT;
    DECLARE OC INT;
    SET OC = ( SELECT id_ordencompra
                FROM ordencompra
                WHERE id_cliente=cliente);
    SET id_cliente = ( SELECT *
                       FROM clientes
                       WHERE cliente=id_cliente);
    SET resultado = ( SELECT SUM(total) cliente
                       FROM ordencompra_info
                       WHERE id_ordencomprainfo= OC);
    RETURN resultado;
END
$$
```

la funcion precio_iva espera como parametros el precio de un producto y el porcentaje que se le va a aplicar al producto como impuesto

```
USE empresa;
```

```
DELIMITER $$
```

```
CREATE FUNCTION precio_iva(precio DECIMAL(11,2),iva INT)
RETURNS DECIMAL(11,2)
NO SQL
BEGIN
    DECLARE resultado DECIMAL(11,2);

    SET resultado = precio * (iva/100);
    RETURN resultado;
END
$$
```

Store procedures

el store procedure sp_clientes_orden espera como parametro una columna de su tabla para ordenarlo y tambien de que manera ordenarlos, si de forma ascendente o descendente

```
USE empresa;
DELIMITER $$;
CREATE PROCEDURE `sp_clientes_orden` (IN field VARCHAR(25), IN orden
VARCHAR(4))
BEGIN
    IF field <> " THEN
        SET @clientes_orden = CONCAT (' ORDER BY',field);
    ELSE
        SET @clientes_orden = "";
    END IF;

    IF orden == "DESC" THEN
        SET @orden = CONCAT(@clientes_orden, orden);

    ELSE
        SET @orden = CONCAT(@clientes_orden,"");
    END IF;

    SET @consulta = CONCAT('SELECT * FROM empresa.clientes',@orden);

    PREPARE var FROM @consulta;
    EXECUTE var;
    DEALLOCATE PREPARE var;
END$$

DELIMITER ;
```

Triggers

El trigger tr_nuevosprecios se activa cada vez que se modifica el precio o el costo del producto creando una nueva tabla donde se almacena el precio o el costo anterior del producto

```
CREATE TRIGGER `tr_nuevosprecios`  
AFTER INSERT ON `productos`  
FOR EACH ROW  
INSERT INTO `precioanterior` (costo,precio_unidad)  
VALUES (NEW.costo,NEW.precio_unidad)
```