

Trabajo Practico Subconsultas – RESTAURANTE.

Dado el siguiente MR de un Restaurante, donde las PK están en negrita y las FK subrayadas, realice las operaciones indicadas

Localidad (**IdLocalidad**, Descripcion)

Plato (**idPlato**, Descripcion, Precio)

Cliente (**IdCliente**, Nombre, Apellido, Calle, Nro, IdLocalidad)

PedidoEncabezado (**IdPedido**, IdCliente, Fecha)

PedidoDetalle (**IdDetalle**, IdPedido, IdPlato, Cantidad)

Cliente.IdLocalidad -> Localidad.IdLocalidad

PedidoEncabezado.IdCliente -> Cliente.idCliente

PedidoDetalle.IdPedido -> PedidoEncabezado.idPedido

PedidoDetalle.IdPlato -> PedidoEncabezado.idPlato

1- Escribir el script de creación de tabla Pedido Detalle con sus restricciones

```
CREATE TABLE PedidoDetalle (idDetalle INT NOT NULL PRIMARY KEY,  
                             idPedido INT NOT NULL,  
                             idPlato VARCHAR (5) NOT NULL,  
                             Cantidad INT NOT NULL,  
                             CONSTRAINT FK4 FOREIGN KEY (idPedido) references  
PedidoEncabezado (idPedido),  
                             CONSTRAINT FK5 FOREIGN KEY (idPlato) references PLATO (idPlato));
```

2- Obtener los datos de todos los clientes, ordenados por Localidad, Nombre y Apellido

```
SELECT *  
FROM CLIENTE c  
INNER JOIN LOCALIDAD L  
ON c.idlocalidad = l.id  
ORDER BY L.Descripcion, Nombre, Apellido
```

3-Informar: número de Pedido, Cantidad de Platos Distintos, Cantidad de unidades total, Importe total del pedido

```
SELECT PD.idpedido, COUNT (DISTINCT pd.IDplato), SUM (Cantidad), SUM (precio*cantidad)  
FROM PedidoDetalle pd  
INNER JOIN PLATO P  
ON pd.idplato = p.Idplato  
GROUP BY idpedido
```

4- Mostrar un detalle de los clientes que han realizado pedidos en el mes de Enero y no realizaron ningún pedido en el mes de marzo

```
SELECT *  
FROM CLIENTE c
```

```
WHERE exists (SELECT PE.IDCLIENTE FROM PEDIDOENCABEZADO pe
              WHERE pe.Fecha BETWEEN '20210101' AND '20210131'
              AND PE.IDcliente = c.idcliente)
AND NOT EXISTS (SELECT PE.IDCLIENTE FROM PEDIDOENCABEZADO pe
                WHERE pe.Fecha BETWEEN '20210301' AND '20210331'
                AND PE.IDcliente = c.idcliente)
```

5 Informar el nombre del plato mas barato de la carta

```
SELECT *
FROM PLATO
WHERE precio = (SELECT MIN(PRECIO) FROM PLATO)
```

6 informar los datos completos de los clientes, la fecha de última compra y el total gastado.

Deben informarse la totalidad de los clientes existentes.

```
SELECT c.idCliente, nombre, apellido, ISNULL( max(fecha),0) AS ultima_Compra, ISNULL (SUM(PD.cantidad *
P.PRECIO), 0)
FROM CLIENTE C
LEFT OUTER JOIN PedidoEncabezado PE
ON C.idcliente = PE.idcliente
LEFT OUTER JOIN Pedidodetalle PD
ON Pe.idPedido = PD.IdPedido
LEFT OUTER JOIN PLATO P
ON PD.IDPLATO = P.IDPLATO
GROUP BY c.idCliente, nombre, apellido
```

7 Informar los platos que han sido comprados por mas de un cliente

```
SELECT *
FROM PLATO P
WHERE idplato IN (SELECT PD.idPlato
                  FROM PedidoEncabezado PE
                  INNER JOIN PedidoDetalle PD ON PE.IDPEDIDO = PD.IDPEDIDO
                  GROUP BY PD.idPlato
                  HAVING COUNT (DISTINCT IDCLIENTE) > 1)
```

8 Mostrar los clientes que han pedido todos los platos del menú

[illegible]

Tambien se puede resolver con COUNT, cuando la cantidad de platos distintos comprados es igual a la cantidad de platos

9- Informar la descripción y precio de los platos que no han sido comprados por ningún cliente.

```
SELECT descripcion, precio
FROM PLATO P
WHERE NOT EXISTS (SELECT 1
                  FROM PedidoDetalle pd
                  WHERE pd.idPlato = p.idPlato)
```

-- Creo tablas para poder usar datos y comprobar

Falta la tabla de PedidoDetalle que deben hacerla ustedes para el punto 1. Sin embargo, sí está agregado un insert con datos para esa tabla, chequen el orden de los atributos en caso que deban modificarlos una vez que creen la tabla.

```
CREATE TABLE Localidad(  
  IdLocalidad INT PRIMARY KEY AUTO_INCREMENT,  
  Descripcion VARCHAR(30) NOT NULL);
```

```
CREATE TABLE Plato (  
  IdPlato INT PRIMARY KEY AUTO_INCREMENT,  
  Descripcion VARCHAR(50) NOT NULL,  
  Precio DOUBLE NOT NULL);
```

```
CREATE TABLE Cliente(  
  IdCliente INT PRIMARY KEY AUTO_INCREMENT,  
  Nombre VARCHAR(20) NOT NULL,  
  Apellido VARCHAR(20) NOT NULL,  
  Calle VARCHAR(50) NOT NULL,  
  Nro INT NOT NULL,  
  IdLocalidad INT NOT NULL,  
  FOREIGN KEY (IdLocalidad) REFERENCES Localidad(IdLocalidad));
```

```
CREATE TABLE PedidoEncabezado(  
  IdPedido INT PRIMARY KEY AUTO_INCREMENT,  
  IdCliente INT NOT NULL,  
  Fecha DATE NOT NULL,  
  FOREIGN KEY (IdCliente) REFERENCES Cliente(IdCliente));
```

-- Agrego datos

```
INSERT INTO Localidad(IdLocalidad, Descripcion)  
VALUES  
(1, "Ituzaingó"),  
(2, "Merlo"),  
(3, "Hurlingham"),  
(4, "Morón");
```

```
INSERT INTO Plato (IdPlato, Descripcion, Precio)  
VALUES  
(1, "Ñoquis con salsa rosa", 523.50),  
(2, "Spaghettis con salsa bolognesa", 572.00),  
(3, "Milanesa con puré", 530.00),  
(4, "Risotto", 575.50),  
(5, "Sopa de calabaza", 480.00),  
(6, "Estofado de carne", 590.50),  
(7, "Ravioles con salsa blanca", 490.00);
```

```
INSERT INTO Cliente(IdCliente, Nombre, Apellido, Calle, Nro, IdLocalidad)
```

VALUES

(1, "Juan", "Perez", "Av. San Miguel", 1740, 2),
(2, "Lucía", "Martinez", "Portugal", 2135, 1),
(3, "Ines", "Gonzalez", "Gobernador Arias", 3567, 4),
(4, "Darío", "Frei", "Av. Vergara", 3216, 3),
(5, "Hector", "Fuentes", "Av. Ratti", 135, 1);

INSERT INTO PedidoEncabezado (IdPedido, IdCliente, Fecha)

VALUES

(1, 2, '2024-01-15'),
(2, 5, '2024-01-25'),
(3, 1, '2024-04-12'),
(4, 3, '2024-02-29'),
(5, 4, '2024-01-06'),
(6, 2, '2024-03-22');

INSERT INTO PedidoDetalle(IdDetalle, Cantidad, IdPedido, IdPlato)

VALUES

(1, 2, 1, 4),
(2, 1, 1, 7),
(3, 3, 2, 3),
(4, 1, 3, 2),
(5, 1, 3, 5),
(6, 1, 3, 6),
(7, 1, 3, 7),
(8, 1, 3, 1),
(9, 1, 3, 3),
(10, 1, 3, 4),
(11, 3, 4, 1),
(12, 2, 5, 6),
(13, 3, 6, 7);