

Bases de Datos Prof. Jefferson Peña

J. Diego Cárdenas Mejía 2416437-3743

Informe Técnico Gastro DB:

Sistema de facturación de restaurante

1. Restricciones empleadas:

En *platillo* y *bebida* se definió un CHECK sobre *importe* en ambas relaciones para impedir la presencia de platillos o bebidas con importes negativos.

En *mesa* se realiza un CHECK sobre *num_Comensales* para garantizar que el valor del atributo sea mayor a 0.

En *factura* se estableció el comportamiento ON DELETE SET NULL para todas las llaves foráneas, a fin de conservar el registro de las facturas emitidas aún cuándo se elimine algún registro al que hace referencia.

2. Resultado consultas:

 Obtener el nombre y apellido de los clientes que hayan consumido un platillo específico:

Se selecciona el nombre y apellido de clientes, haciendo JOIN con factura por cliente_id, dónde factura.id_platillo sea un valor en particular, en este caso se seleccionó el platillo con id_platillo = 27:



2. Obtener el nombre y apellido de los clientes que hayan consumido arroz a la marinera:

De igual manera se seleccionan los nombres y apellidos de cliente, pero ahora se hace JOIN con factura y con platillo, para seleccionar las tuplas dónde platillo.nombre = 'Arroz a la marinera':

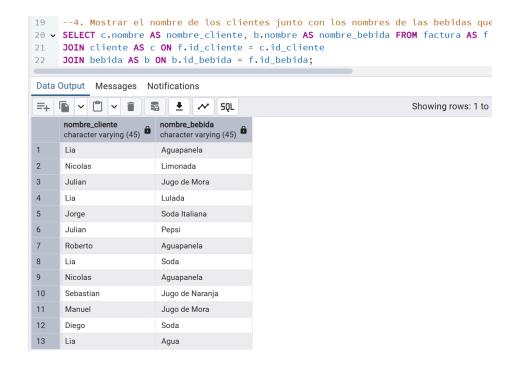


3. Listar el nombre del mesero y la fecha en la que atendió una mesa 10 que se encuentra ubicada en el segundo piso del restaurante:
 Se selecciona el nombre del mesero y fecha_factura, haciendo JOIN con factura y con mesa para seleccionar las tuplas con mesa.id mesa = 10.



4. Mostrar el nombre de los clientes junto con los nombres de las bebidas que consumieron en sus facturas:

Se selecciona cliente.nombre y bebida.nombre de factura, haciendo JOIN con cliente y bebida sobre los identificadores correspondientes:



5. Consultar todas las facturas que incluyan platillos con un importe mayor a \$300000, incluyendo el nombre del cliente y del platillo:

Se selecciona la factura completa factura.* y el cliente.nombre y platillo.nombre, haciendo JOIN con platillo y cliente, y seleccionando las tuplas dónde platillo.importe > 30000.



6. Listar el total de consumo (importe de platillos y bebidas) del cliente llamado Manuel Pedroza Gonzalez:

Se selecciona la suma de platillo.importe y bebida.importe de factura, haciendo un JOIN con cliente, platillo y bebida, y buscando las tuplas con cliente.nombre = 'Manuel' y cliente.apellido = 'Pedroza Gonzalez'.

```
--6.Listar el total de consumo (importe de platillos y bebidas) de

SELECT SUM(p.importe) + SUM(b.importe) AS total FROM factura AS f

JOIN platillo AS p ON f.id_platillo = p.id_platillo

JOIN bebida AS b ON b.id_bebida = f.id_bebida

JOIN cliente AS c ON f.id_cliente = c.id_cliente

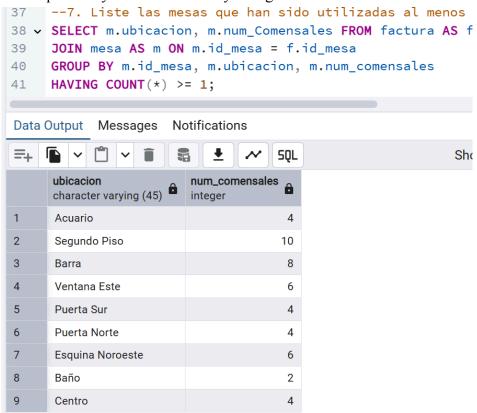
WHERE c.nombre = 'Manuel' AND c.apellido = 'Pedroza Gonzalez';

Data Output Messages Notifications

The value of total bigint and total bigint and total bigint and total bigint and total and
```

7. Liste las mesas que han sido utilizadas al menos una vez, indicando su ubicación y el número de comensales:

Se selecciona la mesa.ubicación y mesa.num_Comensales de factura haciendo JOIN con mesa, agrupando por id_mesa, ubicacion, y num_Comensales, contando y obteniendo aquello cuya cuenta fuese mayor o igual a 1.



3. Prueba restricciones:

- Error al intentar ingresar mesa con *num Comensales* negativo:

```
--Mesa con comensales negativos

NSERT INTO mesa VALUES

Notifications

ERROR: new row for relation "mesa" violates check constraint "chk_num_comensales_positive"

Failing row contains (35, -2, Techo).
```

- Error al ingresar platillo con importe 0:

```
92 --Platillo con importe cero
93 V INSERT INTO platillo VALUES
94 (400, 'Nachos Gratis', 0);
95

Data Output Messages Notifications

ERROR: new row for relation "platillo" violates check constraint "chk_importe_positive"
Failing row contains (400, Nachos Gratis, 0).
```

Valores nulos en factura al eliminar clientes:

