

## 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:

1. *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:

```
4 SELECT cliente.nombre, cliente.apellido FROM factura
5 JOIN cliente ON factura.id_cliente = cliente.id_cliente
6 WHERE factura.id_platillo = 27;
```

Data Output			Messages	Notifications
	nombre character varying (45)	apellido character varying (45)		
1	Diego	Cardenas		
2	Lia	Quijano		
3	Nicolas	Montoya		

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':

```

8  --2.Nombre y apellido de clientes que consumieron arroz a la marinera--
9  v SELECT cliente.nombre, cliente.apellido FROM factura
10 JOIN cliente ON factura.id_cliente = cliente.id_cliente
11 JOIN platillo ON factura.id_platillo = platillo.id_platillo
12 WHERE platillo.nombre = 'Arroz a la marinera';
13

```

	nombre character varying (45)	apellido character varying (45)
1	Nicolas	Montoya
2	Julian	Andrade

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.

```

14  --3. Nombre del mesero y la fecha en la que atendió una mesa 10 que
15  v SELECT m.nombre AS nombre_mesero, f.fecha_factura FROM factura AS f
16 JOIN mesero AS m ON m.id_mesero = f.id_mesero
17 WHERE f.id_mesa = 10;
18

```

	nombre_mesero character varying (45)	fecha_factura date
1	Antonio	2021-06-20
2	Martha	2020-04-13

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:

```

19 --4. Mostrar el nombre de los clientes junto con los nombres de las bebidas que
20 SELECT c.nombre AS nombre_cliente, b.nombre AS nombre_bebida FROM factura AS f
21 JOIN cliente AS c ON f.id_cliente = c.id_cliente
22 JOIN bebida AS b ON b.id_bebida = f.id_bebida;

```

	nombre_cliente character varying (45)	nombre_bebida character varying (45)
1	Lia	Aguapanela
2	Nicolas	Limonada
3	Julian	Jugo de Mora
4	Lia	Lulada
5	Jorge	Soda Italiana
6	Julian	Pepsi
7	Roberto	Aguapanela
8	Lia	Soda
9	Nicolas	Aguapanela
10	Sebastian	Jugo de Naranja
11	Manuel	Jugo de Mora
12	Diego	Soda
13	Lia	Agua

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 donde platillo.importe > 30000.

```

24 --5.Consultar todas las facturas que incluyan platillos con un importe mayor a $300000,
25 SELECT f.*, c.nombre AS nombre_cliente, p.nombre AS nombre_platillo FROM factura AS f
26 JOIN cliente AS c ON f.id_cliente = c.id_cliente
27 JOIN platillo AS p ON f.id_platillo = p.id_platillo
28 WHERE p.importe > 30000;

```

	id_factura integer	fecha_factura date	id_cliente integer	id_mesero integer	id_mesa integer	id_platillo integer	id_bebida integer	nombre_cliente character varying (45)	nombre_platillo character varying (45)
1	52	1992-06-16	8	15	50	26	35	Nicolas	Arroz a la marinera
2	55	1995-05-25	7	17	43	26	34	Julian	Arroz a la marinera
3	57	1995-12-24	2	17	41	23	36	Lia	Garbanzos
4	60	1996-05-15	4	13	50	22	40	Manuel	Frijolada

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'.

```

30 --6. Listar el total de consumo (importe de platillos y bebidas) de
31 v SELECT SUM(p.importe) + SUM(b.importe) AS total FROM factura AS f
32 JOIN platillo AS p ON f.id_platillo = p.id_platillo
33 JOIN bebida AS b ON b.id_bebida = f.id_bebida
34 JOIN cliente AS c ON f.id_cliente = c.id_cliente
35 WHERE c.nombre = 'Manuel' AND c.apellido = 'Pedroza Gonzalez';

```

Data Output		Messages	Notifications
	total bigint		
1	44000		

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.

```

37 --7. Liste las mesas que han sido utilizadas al menos
38 v SELECT m.ubicacion, m.num_Comensales FROM factura AS f
39 JOIN mesa AS m ON m.id_mesa = f.id_mesa
40 GROUP BY m.id_mesa, m.ubicacion, m.num_comensales
41 HAVING COUNT(*) >= 1;

```

Data Output			Messages	Notifications
	ubicacion character varying (45)	num_comensales integer		
1	Acuario	4		
2	Segundo Piso	10		
3	Barra	8		
4	Ventana Este	6		
5	Puerta Sur	4		
6	Puerta Norte	4		
7	Esquina Noroeste	6		
8	Baño	2		
9	Centro	4		

### 3. Prueba restricciones:

- Error al intentar ingresar mesa con num\_Comensales negativo:

```

88 --Mesa con comensales negativos
89 v INSERT INTO mesa VALUES
90 (35, -2, 'Techo');

```

Data Output			Messages	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:

```

96 --Eliminar clientes y revisar factura--
97 DELETE FROM cliente WHERE id_cliente = 1 OR id_cliente = 2;
98 SELECT * FROM factura;

```

Data Output Messages Notifications

Showing rows: 1 to 13

	id_factura integer	fecha_factura date	id_cliente integer	id_mesero integer	id_mesa integer	id_platillo integer	id_bebida integer
1	52	1992-06-16	8	15	50	26	35
2	52	1992-08-20	7	16	49	28	40
3	54	1993-03-20	5	14	44	25	39
4	55	1995-05-25	7	17	43	26	34
5	56	1995-06-21	6	15	46	29	31
6	58	1996-03-14	8	19	48	27	31
7	59	1996-03-21	9	18	46	24	37
8	60	1996-05-15	4	13	50	22	40
9	61	2020-04-13	[null]	16	10	27	36
10	51	1991-01-01	[null]	11	43	27	31
11	53	1993-02-12	[null]	19	45	21	38
12	57	1995-12-24	[null]	17	41	23	36
13	62	2021-06-20	[null]	14	10	24	32