

Tarea 6 - Funciones de agregación

Ejercicio 1

Conteo de frecuencias

```
CREATE TABLE Conteo as
SELECT Nom_art, COUNT(Precio_art)
FROM Nuevo
GROUP BY Nom_art;
```

ABC Nom_art	123 COUNT(Precio_art)
Sandwich	129
Aalopuri	134
Panipuri	150
Vadapav	134
Jugo cania de azucar	153
Cafe frio	161
Frankie	139

Explicación: Seleccionamos el nombre de cada uno de los platillos que se venden en el restaurante, contamos la cantidad que hay por c/precio del platillo, y los agrupamos por sus nombres; al final adjuntamos todo en una tabla llamada: Conteo.

123 MONTH(Fecha)	123 AVG(Monto_trans)
1	275.47619
2	296.666667
3	205.217391
4	270.923077
5	251.845238
6	252.575758
7	263.939394
8	250.411765
9	260.243902
10	253.157895
11	298.591549
12	264.25

Media

```
SELECT MONTH(Fecha), AVG(Monto_trans)
FROM Nuevo
WHERE YEAR(Fecha) = 2022
GROUP BY MONTH(Fecha);
```

Explicación: Seleccionamos los meses de nuestras fechas, calculamos el promedio de ventas que hubo por mes, filtramos que los meses pertenezcan al año 2022, y los agrupamos por sus meses respectivos.

Dificultad: para este ejercicio quise tomar el reto de agrupar por fechas específicas, pero no hallaba la manera de hacerlo.

Hallazgo: encontré las funciones **MONTH(Fecha)** y **YEAR(Fecha)** con los que podemos filtrar un año y un mes en particular.

Ejercicio 2

Máximo

```
CREATE TABLE Maximo AS
SELECT MONTH(Fecha) AS Mes, COUNT(Nom_art) AS Cant_Sandwich
FROM Nuevo
WHERE YEAR(Fecha) = 2023 AND MONTH(Fecha) BETWEEN 01 AND
06 AND Nom_art = 'Sandwich'
GROUP BY MONTH(Fecha);
```

```
SELECT MAX(Cant_Sandwich) FROM Maximo;
```

123 Mes ▼	123 Cant_Sandwich ▼
1	11
2	7
3	9
4	1
5	2
6	2

123 Max(Cant_Sandwich) ▼
11

Explicación: Seleccionamos los meses de nuestras fechas y contamos la cantidad de c/nombre del platillo, filtramos que los meses pertenezcan al año 2023, que se encuentren entre enero y junio y que el platillo sea un Sándwich, y los agrupamos por sus meses respectivos; adjuntamos todo en una tabla llamada: Maximo; usamos la función **MAX(Cant_Sandwich)** para seleccionar el dato con mayor cantidad de Sándwiches.

Dificultad: Sería mucho más útil si en nuestra visualización también apareciera el mes al que pertenece la cantidad máxima, el problema es que la función **MAX(Cant_Sandwich)** no te permite hacer la selección de ambas columnas.

Implementación

```
SELECT MONTH(Fecha) AS Mes, COUNT(Nom_art) AS Cant_Sandwich
FROM Nuevo
WHERE YEAR(Fecha) = 2023 AND MONTH(Fecha) BETWEEN 01 AND
06 AND Nom_art = 'Sandwich'
GROUP BY MONTH(Fecha)
ORDER BY COUNT(Nom_art) DESC
LIMIT 1;
```

123 Mes ▼	123 Cant_Sandwich ▼
1	11

Explicación: el código es el mismo que tenemos en la tabla Maximo, lo único que cambia es la agregación de ordenar los datos de mayor a menor por la cantidad de Sándwich que haya, y limitamos los datos solamente para el primer renglón.

Mínimo

```
CREATE TABLE Maximo AS
SELECT MONTH(Fecha) AS Mes, COUNT(Nom_art) AS Cant_Sandwich
FROM Nuevo
WHERE YEAR(Fecha) = 2023 AND MONTH(Fecha) BETWEEN 01 AND
06 AND Nom_art = 'Sandwich'
GROUP BY MONTH(Fecha);
```

```
SELECT MIN(Cant_Sandwich) FROM Maximo;
```

123 Mes ▼	123 Cant_Sandwich ▼
1	11
2	7
3	9
4	1
5	2
6	2

123 MIN(Cant_Sandwich) ▼
1

Explicación: la lógica es la misma que para calcular el valor máximo; la única diferencia es que aquí hacemos uso de la función **MIN(Cant_Sandwich)**.

Dificultad: la función **MIN(Cant_Sandwich)** no permite seleccionar ambas columnas; presentamos el mismo problema de la función **MAX(Cant_Sandwich)**

Implementación

```
SELECT MONTH(Fecha) AS Mes, COUNT(Nom_art) AS Cant_Sandwich
FROM Nuevo
WHERE YEAR(Fecha) = 2023 AND MONTH(Fecha) BETWEEN 01 AND
06 AND Nom_art = 'Sandwich'
GROUP BY MONTH(Fecha)
ORDER BY COUNT(Nom_art)
LIMIT 1;
```

123 Mes ▼	123 Cant_Sandwich ▼
4	1

Explicación: el código es el mismo que tenemos en la implementación del valor máximo, lo único que cambia es que el orden de la cantidad de Sándwich va ahora de menor a mayor, no de mayor a menor.

Ejercicio 3

Percentil (Cuartil 1)

```
CREATE TABLE Percentil(  
  Posicion INT PRIMARY KEY AUTO_INCREMENT,  
  Id INT,  
  Nom_art VARCHAR(30) DEFAULT 'Y',  
  Cantidad INT NOT NULL,  
  Monto_trans FLOAT(5, 2) NULL  
);
```

```
CREATE TABLE Ordenamiento1 AS  
SELECT Id, Nom_art, Cantidad, Monto_trans  
FROM Nuevo  
ORDER BY Cantidad;
```

```
INSERT INTO Percentil (Id, Nom_art, Cantidad, Monto_trans)  
SELECT Id, Nom_art, Cantidad, Monto_trans  
FROM Ordenamiento1;
```

```
SELECT (1 * COUNT(Cantidad))/4 as Cuartil_1  
FROM Ordenamiento1;
```

```
SELECT Posicion, Id, Nom_art, Cantidad, Monto_trans  
FROM Percentil  
WHERE Posicion = 250;
```

123 Posicion	123 Id	123 Nom_art	123 Cantidad	123 Monto_trans
1	27	Cafe frio	1	40
2	29	Jugo cania de azucar	1	25
3	43	Aalopuri	1	20
4	49	Sandwich	1	60
5	59	Vadapav	1	20
6	63	Jugo cania de azucar	1	25
7	69	Aalopuri	1	20
8	84	Jugo cania de azucar	1	25
9	86	Panipuri	1	20
10	108	Cafe frio	1	40
11	113	Aalopuri	1	20
12	125	Cafe frio	1	40
13	127	Cafe frio	1	40
14	164	Jugo cania de azucar	1	25
15	186	Jugo cania de azucar	1	25
16	205	Vadapav	1	20
17	229	Sandwich	1	60
18	240	Panipuri	1	20
19	242	Frankie	1	50
20	244	Frankie	1	50
21	248	Frankie	1	50
22	254	Jugo cania de azucar	1	25
23	271	Jugo cania de azucar	1	25
24	295	Jugo cania de azucar	1	25
25	326	Panipuri	1	20
26	330	Aalopuri	1	20
27	355	Frankie	1	50
28	358	Sandwich	1	60
29	360	Jugo cania de azucar	1	25
30	384	Sandwich	1	60
31	430	Sandwich	1	60
32	432	Cafe frio	1	40
33	441	Sandwich	1	60
34	448	Jugo cania de azucar	1	25

123 Id	123 Nom_art	123 Cantidad	123 Monto_trans
27	Cafe frio	1	40
29	Jugo cania de azucar	1	25
43	Aalopuri	1	20
49	Sandwich	1	60
59	Vadapav	1	20
63	Jugo cania de azucar	1	25
69	Aalopuri	1	20
84	Jugo cania de azucar	1	25
86	Panipuri	1	20
108	Cafe frio	1	40
113	Aalopuri	1	20
125	Cafe frio	1	40
127	Cafe frio	1	40
164	Jugo cania de azucar	1	25
186	Jugo cania de azucar	1	25
205	Vadapav	1	20
229	Sandwich	1	60
240	Panipuri	1	20
242	Frankie	1	50
244	Frankie	1	50
248	Frankie	1	50
254	Jugo cania de azucar	1	25
271	Jugo cania de azucar	1	25
295	Jugo cania de azucar	1	25
326	Panipuri	1	20
330	Aalopuri	1	20
355	Frankie	1	50
358	Sandwich	1	60
360	Jugo cania de azucar	1	25
384	Sandwich	1	60
430	Sandwich	1	60
432	Cafe frio	1	40
441	Sandwich	1	60
448	Jugo cania de azucar	1	25

123 Cuartil_1 ▼
250

123 Posicion ▼	123 Id ▼	ABC Nom_art ▼	123 Cantidad ▼	123 Monto_trans ▼
250	754	Sandwich	4	240

Explicación: primero creamos la tabla Percentil que contenga la posición de cada uno de nuestros platillos, su Id, su nombre, su cantidad y su monto de venta; esta tabla está originalmente vacía.

Creamos la segunda tabla Ordenamiento1 que contenga la información Id, nombre, cantidad y monto de venta de los platillos que haya en la tabla Nuevo y los ordenamos de menor a mayor.

Colocamos la información que está en la tabla Ordenamiento1 a la tabla Percentil.

Calculamos la posición del cuartil mediante el conteo de cantidades $[(1 * \text{COUNT}(\text{Cantidad}))/ 4]$ que hay en la tabla Ordenamiento1, y seleccionamos todos los datos de la tabla Percentil en donde la posición sea igual a 250.

Dificultad: se tuvo que manipular información de 1 tabla diferente para poder anexar toda la información en otra.

Hallazgo: encontré la función **INSERT INTO** para hacer el traspaso de información de una tabla a otra.

Ejercicio 4

Moda

```
CREATE TABLE Mujer2 as
SELECT Nom_art, COUNT(GPR) AS Mujer
FROM Nuevo
WHERE GPR = 'Sra.'
GROUP BY Nom_art
ORDER BY COUNT(GPR) DESC;
```

```
SELECT MAX(Mujer) AS Moda FROM Mujer2;
```

ABC Nom_art ▼	123 Mujer ▼
Cafe frio	79
Frankie	76
Jugo cania de azucar	74
Panipuri	73
Vadapav	66
Sandwich	60
Aalopuri	60

123 Moda ▼
79

Explicación: seleccionamos el nombre de los platillos y el conteo de los géneros de nuestra base de datos, filtramos que nuestro género sea únicamente de Sra., los agrupamos por sus respectivos nombres, y los ordenamos de mayor a menor. Usamos la función **MAX(Mujer)** para obtener el número mayor del conteo de mujeres. En este caso, la moda sería los 79 Cafés fríos que pidieron las mujeres.

Dificultad: la función **MAX(Mujer)** no permite seleccionar ambas columnas.

Implementación

```
SELECT Nom_art, COUNT(GPR) AS Mujer
FROM Nuevo
WHERE GPR = 'Sra.'
GROUP BY Nom_art
ORDER BY COUNT(GPR) DESC
LIMIT 1;
```

ABC Nom_art ▼	123 Mujer ▼
Cafe frio	79

Explicación: el código es el mismo que tenemos en la tabla Mujer2, lo único que cambia es la agregación de la limitación a solamente el primer renglón.