



ANÁLISIS Y DISEÑO LÓGICO DE SISTEMAS

Noviembre 2023

Luis E.Canales C.
lcanales@utalca.cl

agregados + Join

- Unir estas dos tablas para determinar quién conduce un auto fabricado antes de 2017

```
SELECT...  
FROM Sueldos s, Registros r  
WHERE ...;
```

Sueldos

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000

Registros

UserID	Car	Year
123	Charger	2009
567	Civic	2016
567	Ferrari	2000
789	Camry	2018

agregados + Join

- Uso conjunto de JOIN y agregados
- Problema de testimonio: join y agregar

argmax

agregados + Join

- ¿Cuántos autos fabricados antes del 2017 conduce cada persona?

Sueldos

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000

Registros

UserID	Car	Year
123	Charger	2009
567	Civic	2016
567	Ferrari	2000
789	Camry	2018

agregados + Join

- ¿Cuántos autos fabricados antes del 2017 conduce cada persona?

↑
Agregado: probablemente un **COUNT**, posiblemente también un **GROUP BY**

↖ ↗
Join: Atributos de dos tablas diferentes

Sueldos

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000

Registros

UserID	Car	Year
123	Charger	2009
567	Civic	2016
567	Ferrari	2000
789	Camry	2018

agregados + Join

FWGHOS

- ¿Cuántos autos fabricados antes del 2017 conduce cada persona?

Paso 1: pensar en un Join

```
SELECT ...  
FROM Sueldos s, Registros r  
WHERE s.UserID = r.UserID;
```

La sintaxis implícita de los inner join hace que parezca que las Join ocurren durante la fase WHERE; es un común malentendido

Sueldos

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000

Registros

UserID	Car	Year
123	Charger	2009
567	Civic	2016
567	Ferrari	2000
789	Camry	2018

- ¿Cuántos autos fabricados antes del 2017 conduce cada persona?

Paso 1: pensar en un Join

```
SELECT...  
FROM Sueldos s, Registros r  
WHERE s.UserID = r.UserID;
```

s.UserID	s.Nombre	s.Trabajo	s.Salario	r.UserID	r.Car	r.Year
123	Juan	Contador	500000	123	Charger	2009
567	Magda	Profesora	1200000	567	Civic	2016
567	Magda	Profesora	1200000	567	Ferrari	2000
789	Diana	Profesora	1000000	789	Camry	2018

- ¿Cuántos autos fabricados antes del 2017 conduce cada persona?

Paso 1: pensar en un Join... y el Where

```
SELECT...  
FROM Sueldos s, Registros r  
WHERE s.UserID = r.UserID AND  
        r.Year < 2017;
```

s.UserID	s.Nombre	s.Trabajo	s.Salario	r.UserID	r.Car	r.Year
123	Juan	Contador	500000	123	Charger	2009
567	Magda	Profesora	1200000	567	Civic	2016
567	Magda	Profesora	1200000	567	Ferrari	2000

- ¿Cuántos autos fabricados antes del 2017 conduce cada persona?

Paso 1: pensar en un Join

Paso 2: Group – By usando la clave del Join

```
SELECT s.UserID, COUNT(*)  
FROM Sueldos s, Registros r  
WHERE s.UserID = r.UserID AND  
        r.Year < 2017  
GROUP BY s.UserID;
```

s.UserID	s.Nombre	s.Trabajo	s.Salario	r.UserID	r.Car	r.Year
123	Juan	Contador	500000	123	Charger	2009
567	Magda	Profesora	1200000	567	Civic	2016
567	Magda	Profesora	1200000	567	Ferrari	2000

- ¿Cuántos autos fabricados antes del 2017 conduce cada persona?

Paso 1: pensar en un Join...and where

Paso 2: Group – By usando la clave del Join

```
SELECT s.nombre, COUNT(*)  
FROM Sueldos s, Registros r  
WHERE s.UserID = r.UserID AND  
      r.Year < 2017  
GROUP BY s.UserID, s.nombre;
```

s.UserID	s.Nombre	s.Trabajo	s.Salario	r.UserID	r.Car	r.Year
123	Juan	Contador	500000	123	Charger	2009
567	Magda	Profesora	1200000	567	Civic	2016
567	Magda	Profesora	1200000	567	Ferrari	2000

- ¿Cuántos autos fabricados antes del 2017 conduce cada persona?

Paso 1: pensar en un Join

Paso 2: Group – By usando la clave del Join

```
SELECT s.nombre, COUNT(*) AS Cant
FROM Sueldos s, Registros r
WHERE s.UserID = r.UserID AND
        r.Year < 2017
GROUP BY s.UserID, s.nombre;
```

s.nombre	Cant
Juan	1
Magda	2

agregados + Join

- ¿En qué momento de FWGHOS:
 - ¿Desaparece la tupla de **Aline**?
 - ¿Desaparece la tupla de **Diana**?

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000

UserID	Car	Year
123	Charger	2009
567	Civic	2016
567	Ferrari	2000
789	Camry	2018

s.nombre	Cant
Juan	1
Magda	2

```
SELECT s.nombre, COUNT(*) AS Cant
FROM Sueldos s, Registros r
WHERE s.UserID = r.UserID AND
      r.Year < 2017
GROUP BY s.UserID, s.nombre;
```

- ¿En qué momento de FWGHOS:
 - Algunas personas no tienen auto (Aline)
 - Algunas personas sólo tienen un auto nuevo (Diana)

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000

UserID	Car	Year
123	Charger	2009
567	Civic	2016
567	Ferrari	2000
789	Camry	2018

COUNT(*) nunca será 0 para los grupos ¿Por qué?

s.nombre	Cant
Juan	1
Magda	2

```
SELECT s.nombre, COUNT(*) AS Cant
FROM Sueldos s, Registros r
WHERE s.UserID = r.UserID AND
      r.Year < 2017
GROUP BY s.UserID, s.nombre;
```

agregados vacíos + Joins: Caso #1

FWGHOS

- Pongamos a Aline en el resultado
 - Ignoraremos temporalmente el problema de "mayor que 2017"

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000

UserID	Car	Year
123	Charger	2009
567	Civic	2016
567	Ferrari	2000
789	Camry	2018

```
SELECT s.nombre, COUNT(*) AS Cant
FROM Sueldos s, Registros r
WHERE s.UserID = r.UserID AND
      r.Year < 2017
GROUP BY s.UserID, s.nombre;
```

Caso # 1: Incluyendo grupos vacíos

FWGHOS

- Cómo asegurar que Aline está en el post-join, antes de los resultados de la agregación?

s.UserID	s.Nombre	s.Trabajo	s.Salario	r.UserID	r.Car	r.Year
345	Aline	Contador	6000000	??	??	??
123	Juan	Contador	500000	123	Charger	2009
567	Magda	Profesora	900000	567	Civic	2016
567	Magda	Profesora	900000	567	Ferrari	2000
789	Diana	Profesora	1000000	789	Camry	2018

```
SELECT s.nombre, COUNT(*) AS Cant
FROM Sueldos s, Registros r
WHERE s.UserID = r.UserID
GROUP BY s.UserID, s.nombre;
```

Caso # 1: Incluyendo grupos vacíos

FWGHOS

- Cómo asegurar que Aline está en el resultado del post join?
 - Un outer join!

s.UserID	s.Nombre	s.Trabajo	s.Salario	r.UserID	r.Car	r.Year
345	Aline	Contador	6000000	NULL	NULL	NULL
123	Juan	Contador	500000	123	Charger	2009
567	Magda	Profesora	900000	567	Civic	2016
567	Magda	Profesora	900000	567	Ferrari	2000
789	Diana	Profesora	1000000	789	Camry	2018

s.nombre	Cant
Aline	1
Juan	1
Magda	2
Diana	1

```
SELECT s.nombre, COUNT(*) AS Cant
FROM Sueldos s, LEFT OUTER JOIN Registros r
ON s.UserID = r.UserID
GROUP BY s.UserID, s.nombre;
```


Caso # 1: Incluyendo grupos vacíos

FWGHOS

- ¿Cómo asegurarse de que el **count** de Aline es 0?
 - ¡Contar los valores no NULL!

p.UserID	p.Nombre	p.Trabajo	p.Salario	r.UserID	r.Car	r.Year
345	Aline	Contador	6000000	NULL	NULL	NULL
123	Juan	Contador	500000	123	Charger	2009
567	Magda	Profesora	900000			2016
567	Magda	Profesora	900000			2000
789	Diana	Profesora	1000000			2018

COUNT(attr) no incluye NULL, por lo que puede devolver 0!

p.nombre	Cant
Aline	0
Juan	1
Magda	2
Diana	1

```
SELECT s.nombre, COUNT(r.UserID) AS Cant
FROM Sueldos s, LEFT OUTER JOIN Registros r
ON s.UserID = r.UserID
GROUP BY s.UserID, s.nombre;
```

Caso # 2: Incluyendo grupos vacíos

FWGHOS

- ¿Cómo incluimos los resultados "anteriores a 2017"?
 - Por el momento, ¡no hay nada que hacer! Una forma de hacerlo es utilizando subconsultas

Sueldos

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000

Registros

UserID	Car	Year
123	Charger	2009
567	Civic	2016
567	Ferrari	2000
789	Camry	2018

```
SELECT s.Nombre, COUNT(r.UserID) AS cnt
FROM Sueldos s LEFT OUTER JOIN Registros r
ON s.UserID = r.UserID
GROUP BY s.UserID, s.Nombre;
```

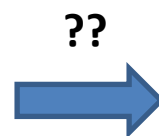
Resumen

- Uso conjunto de Joins y agregados
- Problema de los testigos: Join y agregación

El Problema del Testimonio: Ejemplo

- ¿Qué persona tiene el salario más alto por trabajo?
 - El ejemplo anterior sólo devuelve el salario por puesto de trabajo

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000



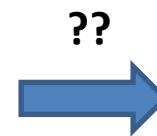
Trabajo	Salario
Contador	600000
Profesora	1200000

El Problema del Testimonio: Una solución

- Grupo anterior por ejemplo:

```
SELECT Trabajo, MAX(Salario)
FROM Sueldos
GROUP BY Trabajo
```

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000



Trabajo	Salario
Contador	600000
Profesora	1200000

El Problema del Testimonio: Una solución

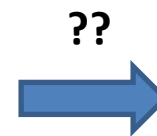
- Grupo anterior por ejemplo:

```
SELECT    Trabajo, MAX(Salario)
FROM      Sueldos
GROUP BY  Trabajo
```

- ¿Qué pasa con esto?

```
SELECT    Nombre, MAX(Salario)
FROM      Sueldos
GROUP BY  Trabajo
```

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000



Trabajo	Salario
Contador	600000
Profesora	1200000

El Problema del Testimonio: Una solución

- Grupo anterior por ejemplo:

```
SELECT    Trabajo, MAX(Salario)
FROM      Sueldos
GROUP BY  Trabajo
```

- ¿Qué pasa con esto?

```
SELECT    Nombre, MAX(Salario)
FROM      Sueldos
GROUP BY  Trabajo
```

Error: debe incluir
Nombre en GROUP BY

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000

??



Trabajo	Salario
Contador	600000
Profesora	1200000

El Problema del Testimonio: Una solución

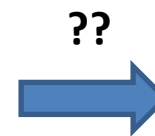
- Grupo anterior por ejemplo:

```
SELECT    Trabajo, MAX(Salario)
FROM      Sueldos
GROUP BY  Trabajo
```

- ¿Mejor?

```
SELECT    Nombre, MAX(Salario)
FROM      Sueldos
GROUP BY  Trabajo, Nombre
```

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000



Trabajo	Salario
Contador	600000
Profesora	1200000

El Problema del Testimonio: Una solución

- Grupo anterior por ejemplo:

```
SELECT Trabajo, MAX(Salario)
FROM Sueldos
GROUP BY Trabajo
```

- ¿Mejor?

```
SELECT Nombre, MAX(Salario)
FROM Sueldos
GROUP BY Trabajo, Nombre
```

No: ahora simplemente
devolver a todos los
empleados...

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000

??

Trabajo	Salario
Contador	600000
Profesora	1200000

El Problema del Testimonio

- Es fácil calcular $\max(\dots)$ en SQL

```
SELECT Trabajo, MAX(Salario)
FROM Sueldos
GROUP BY Trabajo
```

- Es difícil calcular $\operatorname{argmax}(\dots)$, es decir, el testigo
- Gracias a FWGHOS, sabemos que SELECT, HAVING, ORDER BY deben usar cualquiera de los dos:
 - Funciones agregadas, o
 - Columnas en **GROUP BY**

El Problema del Testimonio

- Es fácil calcular $\max(\dots)$ en SQL

```
SELECT Trabajo, MAX(Salario)
FROM Sueldos
GROUP BY Trabajo
```

- Es difícil calcular $\operatorname{argmax}(\dots)$, es decir, el testigo

🤔 Experimento de pensamiento 🤔

Por qué no tiene sentido encontrar el argmax con una función de agregación COUNT o AVG?

El Problema del Testimonio: Idea clave

- "Iterar" una segunda vez sobre la misma tabla
 - Entonces "devuelve" las filas donde **Salario = max(Salario)**
 - Se puede hacer elegantemente en una consulta GROUP BY!

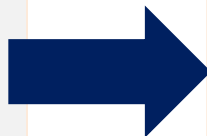
UserID	Nombre	Trabajo	Salario	MaxSal
123	Juan	Contador	500000	600000
345	Aline	Contador	600000	600000
567	Magda	Profesora	1200000	1200000
789	Diana	Profesora	1000000	1200000

El Problema del Testimonio: Idea clave

- "Iterar" una segunda vez sobre la misma tabla
 - Entonces "devuelve" las filas donde **Salario = max(Salario)**
 - Se puede hacer elegantemente en una consulta GROUP BY!

UserID	Nombre	Trabajo	Salario	MaxSal
123	Juan	Contador	500000	600000
345	Aline	Contador	600000	600000
567	Magda	Profesora	1200000	1200000
789	Diana	Profesora	1000000	1200000

```
SELECT Trabajo, MAX(Salario)
FROM Sueldos
GROUP BY Trabajo
```



```
SELECT S1.Nombre, S1.Salario
FROM Sueldos AS S1, Sueldos AS S2
WHERE S1.Trabajo = S2.Trabajo
GROUP BY S1.Nombre, S1.Salario, S2.Trabajo
HAVING S1.Salario = MAX(S2.Salario)
```

El Problema del Testimonio: Paso 1

- Calcular todas las parejas por trabajo

UserID	Nombre	Trabajo	Salario
123	Juan	Contador	500000
345	Aline	Contador	600000
567	Magda	Profesora	1200000
789	Diana	Profesora	1000000



???

S1.Nombre	S1.Trabajo	S1.Salario	S2.Nombre	S2.Trabajo	S2.Salario
Juan	Contador	500000	Juan	Contador	500000
Juan	Contador	500000	Aline	Contador	600000
Juan	Contador	500000	Magda	Profesora	1200000
Juan	Contador	500000	Diana	Profesora	1000000
Aline	Contador	600000	Juan	Contador	500000
Aline	Contador	600000	Aline	Contador	600000
....

El Problema del Testimonio: Paso 1

- "Iterar" una segunda vez sobre la misma tabla
 - Pista: **FWGHOS** (¡todavía no hemos agregado!)

```
SELECT...  
FROM      Sueldos AS S1,  
          Sueldos AS S2  
  
WHERE...  
GROUP BY...  
HAVING...
```

???



???

S1.Nombre	S1.Trabajo	S1.Salario	S2.Nombre	S2.Trabajo	S2.Salario
Juan	Contador	500000	Juan	Contador	500000
Juan	Contador	500000	Aline	Contador	600000
Juan	Contador	500000	Magda	Profesora	1200000
Juan	Contador	500000	Diana	Profesora	1000000
Aline	Contador	600000	Juan	Contador	500000
Aline	Contador	600000	Aline	Contador	600000
....

El Problema del Testimonio: Paso 1

- "Iterar" una segunda vez sobre la misma tabla
 - Pista: FWGHOS (¡todavía no hemos agregado!)

```
SELECT...  
FROM      Sueldos AS S1,  
          Sueldos AS S2  
WHERE     S1.Trabajo = S2.Trabajo  
GROUP BY...  
HAVING...
```



S1.Nombre	S1.Trabajo	S1.Salario	S2.Nombre	S2.Trabajo	S2.Salario
Juan	Contador	500000	Juan	Contador	500000
Juan	Contador	500000	Aline	Contador	600000
Juan	Contador	500000	Magda	Profesora	1200000
Juan	Contador	500000	Diana	Profesora	1000000
Aline	Contador	600000	Juan	Contador	500000
Aline	Contador	600000	Aline	Contador	600000
....

El Problema del Testimonio: Paso 2

- Agregar una columna para la tabla derecha (P2) por trabajo máximo

```
SELECT...  
FROM    Sueldos AS S1,  
        Sueldos AS S2  
WHERE    S1.Trabajo = S2.Trabajo  
GROUP BY...  
HAVING...
```

???



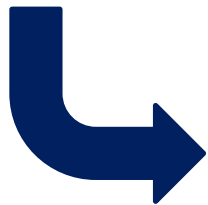
???

S1.Nombre	S1.Trabajo	S1.Salario	S2.Nombre	S2.Trabajo	S2.Salario
Juan	Contador	500000	Juan	Contador	500000
Juan	Contador	500000	Aline	Contador	600000
Aline	Contador	600000	Juan	Contador	500000
Aline	Contador	600000	Aline	Contador	600000
Magda	Profesora	1200000	Magda	Profesora	1200000
Magda	Profesora	1200000	Diana	Profesora	1000000
Diana	Profesora	1000000	Magda	Profesora	1200000
Diana	Profesora	1000000	Diana	Profesora	1000000

El Problema del Testimonio: Paso 2

- Agregar una columna para la tabla derecha (S2) por trabajo máximo

```
SELECT...  
FROM    Sueldos AS S1,  
        Sueldos AS S2  
WHERE    S1.Trabajo = S2.Trabajo  
GROUP BY S2.Trabajo  
HAVING...
```



???

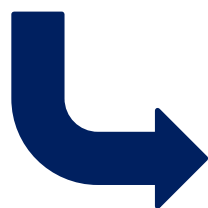
S1.Nombre	S1.Trabajo	S1.Salario	S2.Nombre	S2.Trabajo	S2.Salario	MAX(S2.Salario)
Juan	Contador	500000	Juan	Contador	500000	600000
Juan	Contador	500000	Aline	Contador	600000	600000
Aline	Contador	600000	Juan	Contador	500000	600000
Aline	Contador	600000	Aline	Contador	600000	600000
Magda	Profesora	1200000	Magda	Profesora	1200000	1200000
Magda	Profesora	1200000	Diana	Profesora	1000000	1200000
Diana	Profesora	1000000	Magda	Profesora	1200000	1200000
Diana	Profesora	1000000	Diana	Profesora	1000000	1200000

El Problema del Testimonio: Paso 2

- Agregar una columna para la tabla derecha (P2) por trabajo máximo

```
SELECT... ? MAX(S2.Salario)
FROM      Sueldos AS S1,
          Sueldos AS S2
WHERE     S1.Trabajo = S2.Trabajo
GROUP BY  S2.Trabajo
HAVING... ? MAX(S2.Salario)
```

No estamos seguros de dónde colocar la agregación, pero sabemos que necesitamos calcularla



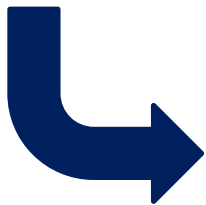
???

S1.Nombre	S1.Trabajo	S1.Salario	S2.Nombre	S2.Trabajo	S2.Salario	MAX(S2.Salario)
Juan	Contador	500000	Juan	Contador	500000	600000
Juan	Contador	500000	Aline	Contador	600000	600000
Aline	Contador	600000	Juan	Contador	500000	600000
Aline	Contador	600000	Aline	Contador	600000	600000
Magda	Profesora	1200000	Magda	Profesora	1200000	1200000
Magda	Profesora	1200000	Diana	Profesora	1000000	1200000
Diana	Profesora	1000000	Magda	Profesora	1200000	1200000
Diana	Profesora	1000000	Diana	Profesora	1000000	1200000

El Problema del Testimonio: Paso 3

- Necesidad de retener:
 - Salario "original" (para comparar con el máximo por puesto de trabajo)
 - Nombre "original" (el problema)

```
SELECT...  
FROM    Sueldos AS S1,  
        Sueldos AS S2  
WHERE   S1.Trabajo = S2.Trabajo  
GROUP BY S1.nombre, S1.Salario, S2.Trabajo  
HAVING...
```



S1.Nombre	S1.Salario			MAX(S2.Salario)
Juan	500000	Juan	500000	600000
Aline	600000	Juan	500000	600000
Magda	1200000	Magda	1200000	1200000
Diana	1000000	Magda	1200000	1200000

El Problema del Testimonio: Paso 4

- Descartar los grupos que no coinciden

```
SELECT...  
FROM    Sueldos AS S1,  
        Sueldos AS S2  
WHERE    S1.Trabajo = S2.Trabajo  
GROUP BY S1.nombre, S1.Salario, S2.Trabajo  
HAVING...
```

???



???

S1.Nombre	S1.Salario	MAX(S2.Salario)
Juan	500000	600000
Aline	600000	600000
Magda	1200000	1200000
Diana	1000000	1200000

El Problema del Testimonio: Paso 4

- Descartar los grupos que no coinciden

```
SELECT...  
FROM    Sueldos AS S1,  
        Sueldos AS S2  
WHERE   S1.Trabajo = S2.Trabajo  
GROUP BY S1.nombre, S1.Salario, S2.Trabajo  
HAVING  S1.Salario = MAX(S2.Salario)
```

¡Aquí está nuestra agregación!



S1.Nombre	S1.Salario	MAX(S2.Salario)
Aline	600000	600000
Magda	1200000	1200000

El Problema del Testimonio: Paso 5

- ¡Seleccione sus campos!

```
SELECT...  
FROM      Sueldos AS S1,  
          Sueldos AS S2  
WHERE     S1.Trabajo = S2.Trabajo  
GROUP BY  S1.nombre, S1.Salario, S2.Trabajo  
HAVING    S1.Salario = MAX(S2.Salario)
```

???



???

S1.Nombre	S1.Salario	MAX(S2.Salario)
Aline	600000	600000
Magda	1200000	1200000

El Problema del Testimonio: Paso 5

- ¡Seleccione sus campos!

```
SELECT  S1.Nombre, S1.Salario
FROM    Sueldos AS S1,
        Sueldos AS S2
WHERE   S1.Trabajo = S2.Trabajo
GROUP BY S1.nombre, S1.Salario, S2.Trabajo
HAVING  S1.Salario = MAX(S2.Salario)
```



S1.Nombre	S1.Salario	MAX(S2.Salario)
Aline	600000	600000
Magda	1200000	1200000

La receta del problema de los testigos

1. Comenzar con un Self Join
2. Identificar el grupo y la agregación por grupo fn
 - Ej: **por puesto de trabajo** **salario máximo**

```
SELECT    S1.Nombre, MAX(S2.Salario)
FROM      Sueldos AS S1,
            Sueldos AS S2
WHERE      S1.Trabajo = S2.Trabajo
GROUP BY  S1.nombre, S1.Salario, S2.Trabajo
HAVING    S1.Salario = MAX(S2.Salario)
```

La receta del problema de los testigos

1. Comenzar con un Self Join
2. Identificar el grupo y la agregación por grupo fn
3. Join a las tablas originales sobre el atributo o atributos de agrupación

```
SELECT    S1.Nombre, MAX(S2.Salario)
FROM      Sueldos AS S1,
            Sueldos AS S2
WHERE      S1.Trabajo = S2.Trabajo
GROUP BY  S1.nombre, S1.Salario, S2.Trabajo
HAVING    S1.Salario = MAX(S2.Salario)
```

La receta del problema de los testigos

1. Comenzar con un Self Join
2. Identificar el grupo y la agregación por grupo fn
3. Join a las tablas originales sobre el atributo o atributos de agrupación
4. También agrupar en el atributo del argumento (original)
 - Ejemplo: el nombre y el salario de la tupla "contribuyente".

```
SELECT    S1.Nombre, MAX(S2.Salario)
FROM      Sueldos AS S1,
            Sueldos AS S2
WHERE      S1.Trabajo = S2.Trabajo
GROUP BY  S1.nombre, S1.Salario, S2.Trabajo
HAVING    S1.Salario = MAX(S2.Salario)
```

La receta del problema de los testigos

1. Comenzar con un Self Join
2. Identificar el grupo y la agregación por grupo fn
3. Join a las tablas originales sobre el atributo o atributos de agrupación
4. También agrupar en el atributo del argumento (original)
5. Eliminar los grupos cuyos valores no coinciden con el nuevo argumento agregado
6. Seleccionar el argumento y su valor agregado
7. ...ganancias?

```
SELECT    S1.Nombre, MAX(S2.Salario)
FROM      Sueldos AS S1,
            Sueldos AS S2
WHERE      S1.Trabajo = S2.Trabajo
GROUP BY  S1.nombre, S1.Salario, S2.Trabajo
HAVING     S1.Salario = MAX(S2.Salario)
```

El problema de los testigos: comparación

- Busquemos los análogos entre:
 - Calcular el Max

```
SELECT Trabajo, MAX(Salario)
FROM Sueldos
GROUP BY Trabajo
```

- Calcular el ARGmax

```
SELECT S1.Nombre, MAX(S2.Salario)
FROM Sueldos AS S1,
      Sueldos AS S2
WHERE S1.Trabajo = S2.Trabajo
GROUP BY S1.nombre, S1.Salario, S2.Trabajo
HAVING S1.Salario = MAX(S2.Salario)
```

El problema de los testigos: comparación

- Busquemos los análogos entre:
 - Calcular el Max

```
SELECT    Trabajo, MAX(Salario)
FROM      Sueldos
GROUP BY  Trabajo
```

- Calcular el ARGmax

```
SELECT    S1.Nombre, S1.Salario
FROM      Sueldos AS S1,
          Sueldos AS S2
WHERE     S1.Trabajo = S2.Trabajo
GROUP BY  S1.nombre, S1.Salario, S2.Trabajo
HAVING    S1.Salario = MAX(S2.Salario)
```

El problema de los testigos: comparación

- Busquemos los análogos entre:
 - Calcular el Max

```
SELECT    Trabajo, MAX(Salario)
FROM      Sueldos
GROUP BY  Trabajo
```

- Calcular el ARGmax

```
SELECT    S1.Nombre, S1.Salario
FROM      Sueldos AS S1,
          Sueldos AS S2
WHERE     S1.Trabajo = S2.Trabajo
GROUP BY  S1.nombre, S1.Salario, S2.Trabajo
HAVING    S1.Salario = MAX(S2.Salario)
```

El problema de los testigos: comparación

- Busquemos los análogos entre:
 - Calcular el Max

```
SELECT Trabajo, MAX(Salario)
FROM Sueldos
GROUP BY Trabajo
```

- Calcular el ARGmax

```
SELECT S1.Nombre, S1.Salario
FROM Sueldos AS S1,
      Sueldos AS S2
WHERE S1.Trabajo = S2.Trabajo
GROUP BY S1.nombre, S1.Salario, S2.Trabajo
HAVING S1.Salario = MAX(S2.Salario)
```


El problema de los testigos: comparación

- Vamos a buscar al testigo:
 - Calcular el Max

```
SELECT Trabajo, MAX(Salario)
FROM Sueldos
GROUP BY Trabajo
```

- Calcular el ARGmax

```
SELECT S1.Nombre, S1.Salario
FROM Sueldos AS S1,
      Sueldos AS S2
WHERE S1.Trabajo = S2.Trabajo
GROUP BY S1.nombre, S1.Salario, S2.Trabajo
HAVING S1.Salario = MAX(S2.Salario)
```

El problema de los testigos: comparación

- Vamos a buscar al testigo:
 - Calcular el Max

```
SELECT Trabajo, MAX(Salario)
FROM Sueldos
GROUP BY Trabajo
```

- Calcular el ARGmax

```
SELECT S1.Nombre, S1.Salario
FROM Sueldos AS S1,
      Sueldos AS S2
WHERE S1.Trabajo = S2.Trabajo
GROUP BY S1.nombre, S1.Salario, S2.Trabajo
HAVING S1.Salario = MAX(S2.Salario)
```

El problema de los testigos: comparación

- Vamos a buscar al testigo:
 - Calcular el Max

```
SELECT Trabajo, MAX(Salario)
FROM Sueldos
GROUP BY Trabajo
```

- Calcular el ARGmax

```
SELECT S1.Nombre, S1.Salario
FROM Sueldos AS S1,
      Sueldos AS S2
WHERE S1.Trabajo = S2.Trabajo
GROUP BY S1.nombre, S1.Salario, S2.Trabajo
HAVING S1.Salario = MAX(S2.Salario)
```

El problema de los testigos: reflexiones finales

- Argmax, o la búsqueda del testigo, es una consulta común en SQL
- Hemos visto cómo escribirla en una única consulta SELECT FROM WHERE GROUP BY con un self join
 - También se puede hacer con una consulta anidada