Carrera profesional
Computación

CONSULTAS – Recuperación de Datos
Parte 2

Mg. Alex Tito Belleza Porras

RECUPERACIÓN DE DATOS

Tema : Agrupamiento de datos

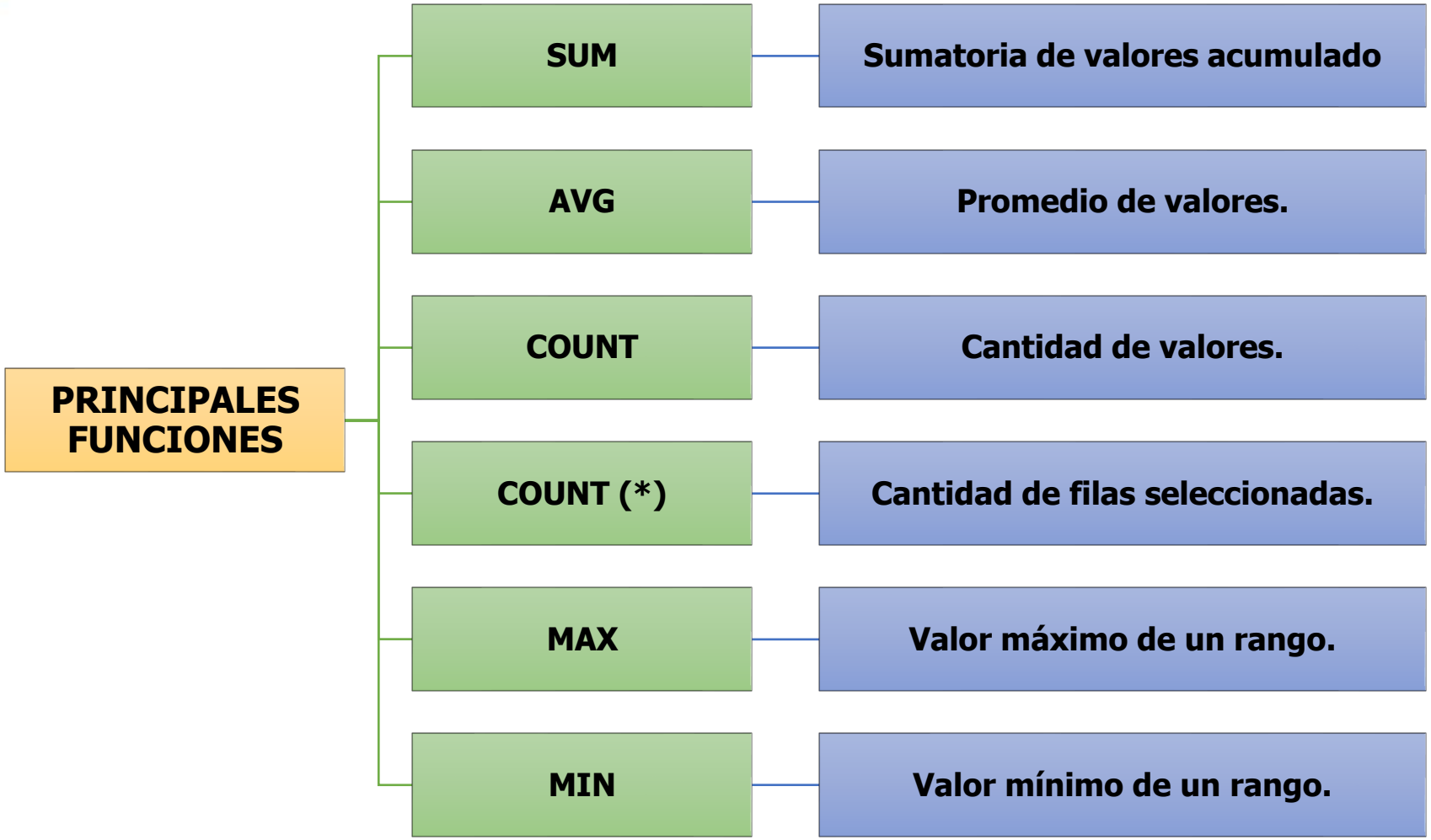
- **Empleo de funciones agregadas**
- **Empleo de GROUP BY, HAVING**

AGRUPAMIENTO DE DATOS

EMPEZAREMOS MENCIONANDO

FUNCIONES AGREGADAS

¿CUÁLES SON LAS FUNCIONES AGREGADAS?



ejemplos

FUNCIONES AGREGADAS

Nota: Utilizar la base de datos **NEGOCIOS**

1.) SUM

Ejemplo: Mostrar la sumatoria total del unidades en existencia de la tabla productos:

```
SELECT  SUM(UNIDADESENEXISTENCIA) AS [TOTAL STOCK ACTUAL]
FROM COMPRAS.PRODUCTOS
GO
```

TOTAL STOCK ACTUAL
3119

2.) AVG

Ejemplo: Calcular la edad promedio de los empleados.

```
SELECT  AVG(DATEDIFF(YY, FECNAC, GETDATE()))AS EDADPROMEDIO
FROM RRHH.EMPLEADOS
GO
```

EDADPROMEDIO
55

3.) COUNT

Ejemplo: Mostrar la cantidad total de Proveedores

```
SELECT  COUNT(P.IDPROVEEDOR) AS TOTAL_PROVEEDORES
FROM COMPRAS.PROVEEDORES AS P
GO
```

TOTAL_PROVEEDORES
29

4.) MIN
5.) MAX

Ejemplo: **Mostrar Precioxunidad mínimo y el máximo de los productos.**

```
SELECT MAX(P.PRECIOUNIDAD)
AS MAXIMOPRECIOUNIDAD
FROM COMPRAS.PRODUCTOS AS P
```

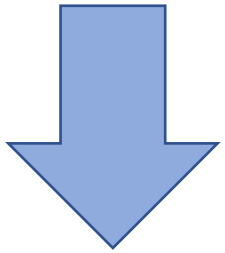
MAXIMOPRECIOUNIDAD
263

```
SELECT
MIN(P.PRECIOUNIDAD) AS
ELMINIMOPRECIOUNIDAD,
MAX(P.PRECIOUNIDAD) AS
MAXIMOPRECIOUNIDAD
FROM COMPRAS.PRODUCTOS AS P
```

ELMINIMOPRECIOUNIDAD	MAXIMOPRECIOUNIDAD
2	263

AGRUPAMIENTO DE DATOS

AHORA TRATAREMOS



GROUP BY y HAVING

Imagen 01

Observa y analiza

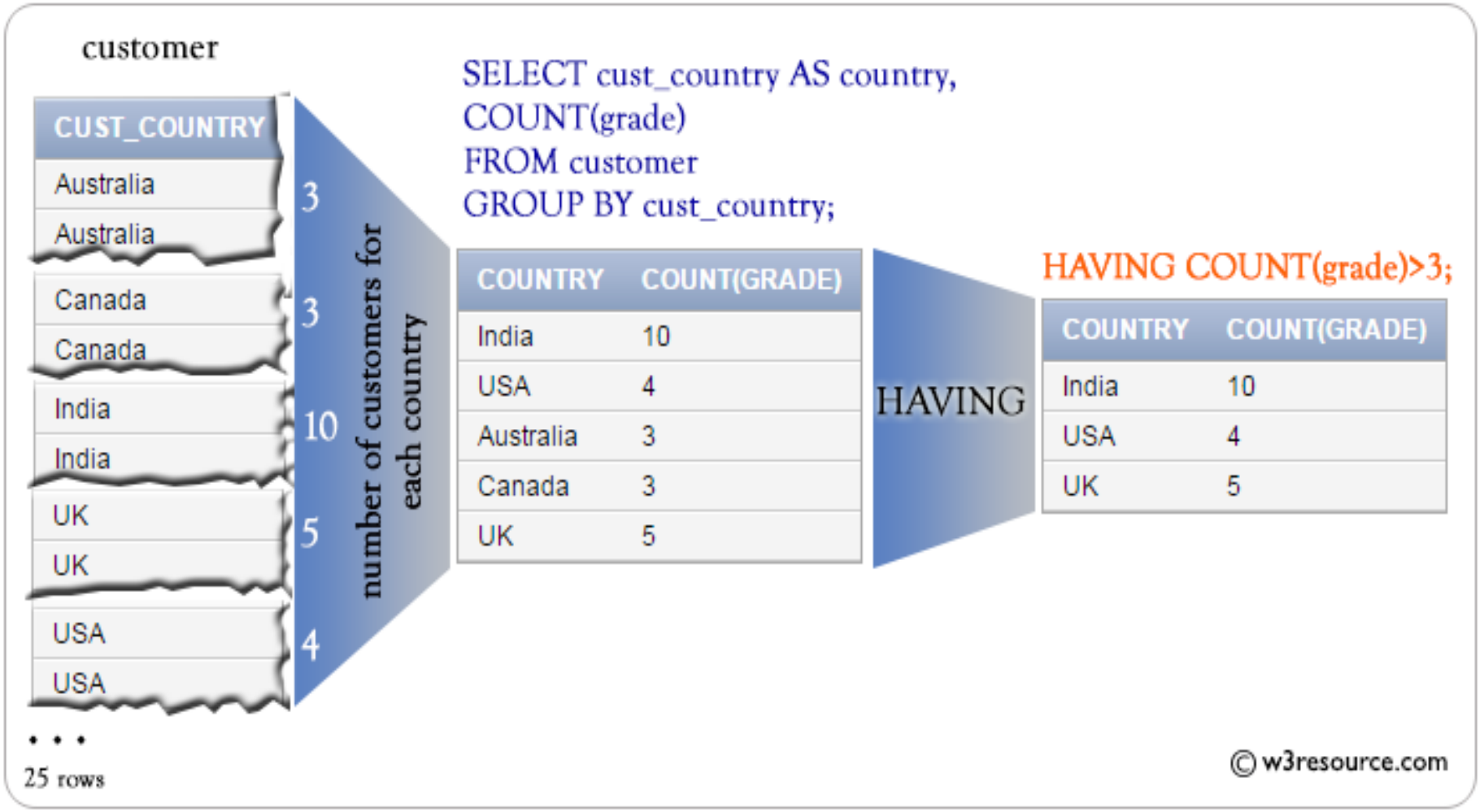
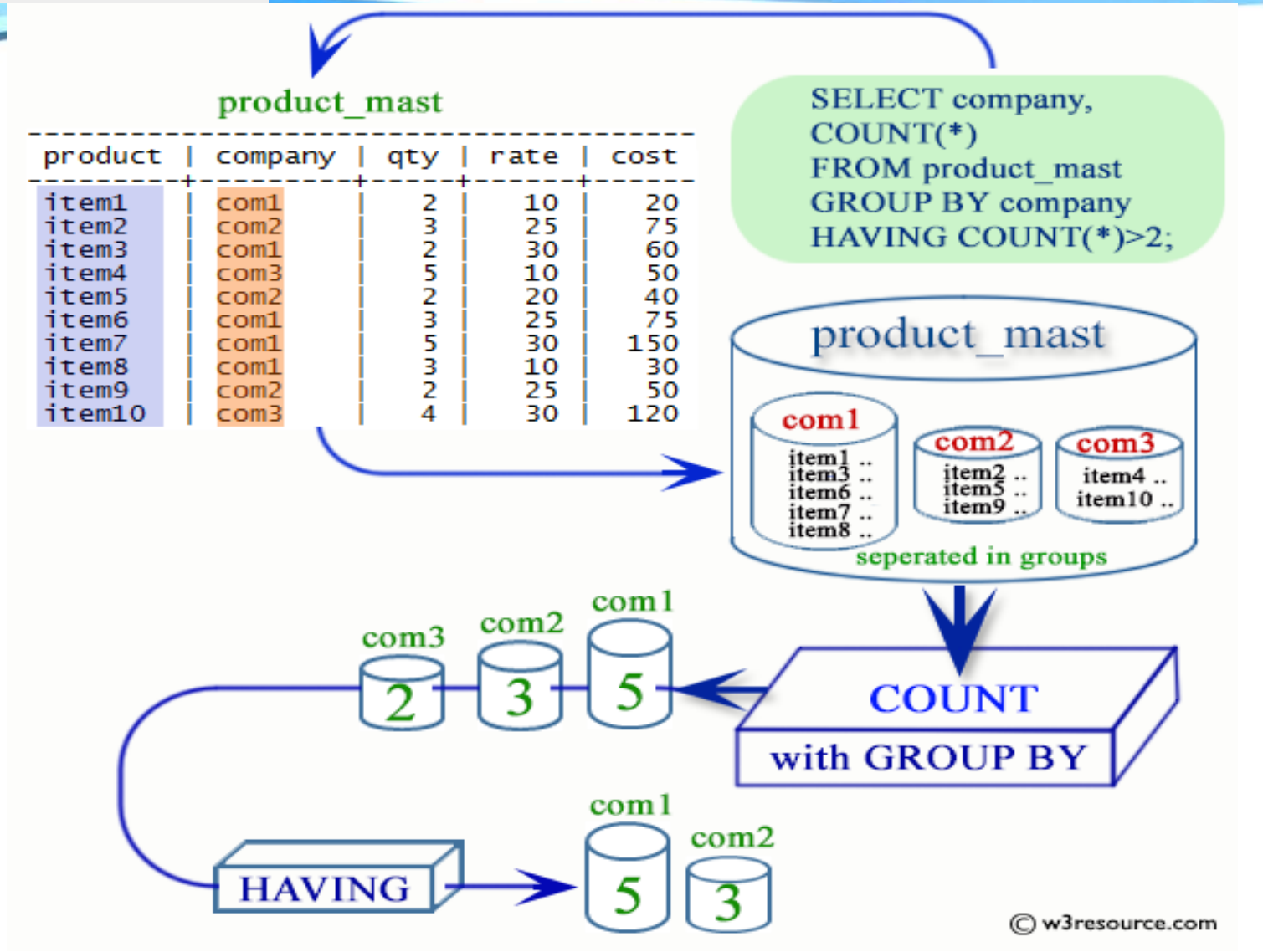


Imagen 02

Observa y analiza



¿GROUP BY y HAVING?

Si deseas generar **valores resumidos** para una columna.

La solución es con **GROUP BY**

Ya que, combina registros con valores idénticos, en la lista de campos específicos, en un único registro.

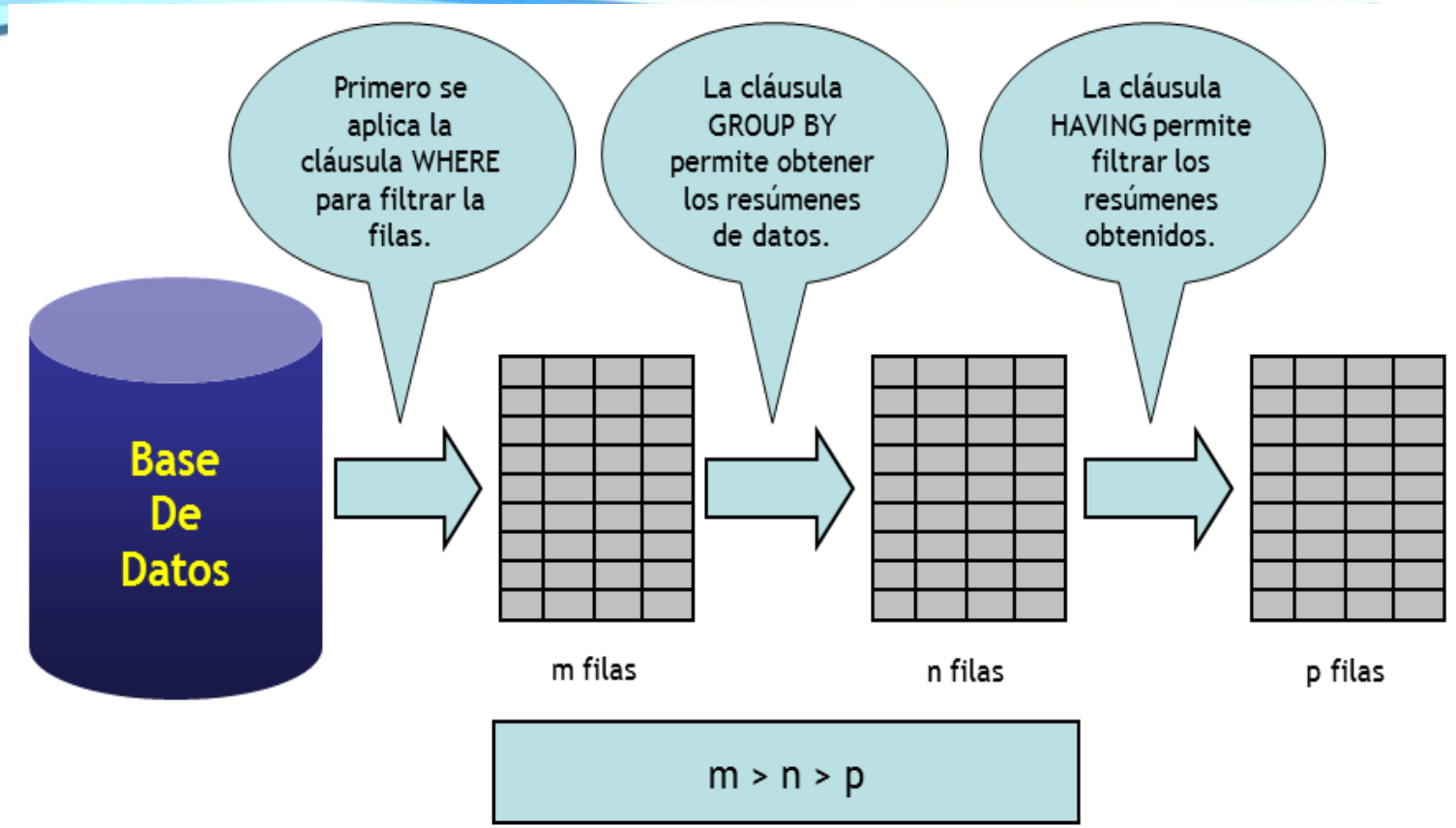
¿Y HAVING?

La cláusula **HAVING** es opcional y se usa sólo con la cláusula **GROUP BY** para **restringir los grupos de filas** que son presentadas en un resultado

Nota

Los valores **NULL** de los campos **GROUP BY se agrupan** y no se omiten

AGRUPANDO DATOS



Enlace http://gcoronelc.blogspot.com/2013/06/sql-server-agrupando-datos.html

Vídeo https://www.youtube.com/watch?v=qrpIOiKOAzE

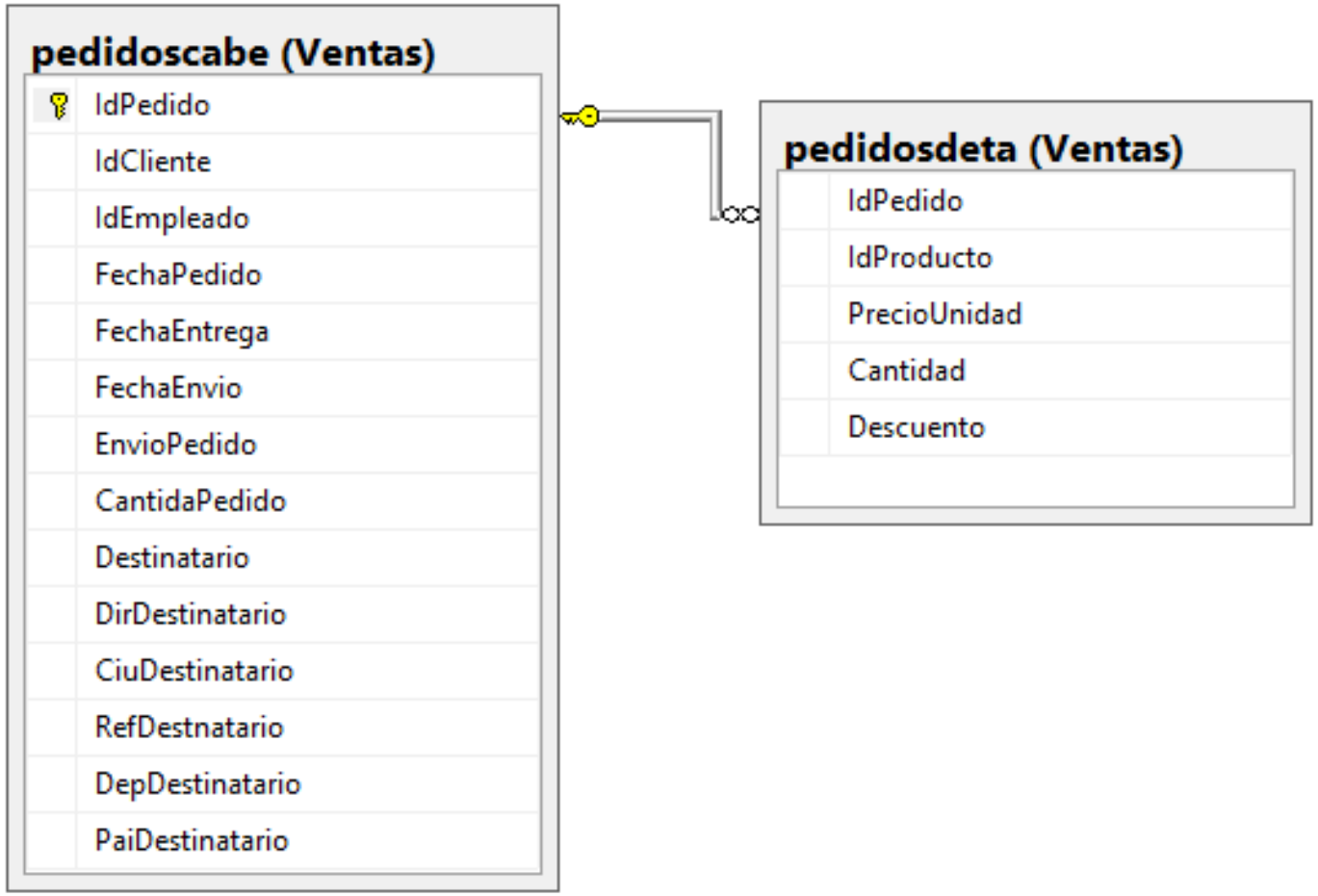
ejemplos

GROUP y HAVING

Nota: Utilizar la base de datos
NEGOCIOS y DEPARTAMENTOS

Ejemplo 01

Utilizando las tablas **Ventas.pedidoscabe** y **Ventas.pedidosdeta**



Ejemplo 01

1

IDPEDIDO	FECHAPEDIDO	CANTIDAD	PRECIOUNIDAD
10248	1996-07-04 00:00:00.000	12	14
10248	1996-07-04 00:00:00.000	10	9
10248	1996-07-04 00:00:00.000	5	34
10249	1996-07-05 00:00:00.000	9	18
10249	1996-07-05 00:00:00.000	40	42
10250	1996-07-08 00:00:00.000	10	7
10250	1996-07-08 00:00:00.000	35	42
10250	1996-07-08 00:00:00.000	15	16
10251	1996-07-08 00:00:00.000	6	16
10251	1996-07-08 00:00:00.000	15	15
10251	1996-07-08 00:00:00.000	20	16
10252	1996-07-09 00:00:00.000	40	64
10252	1996-07-09 00:00:00.000	25	2
10252	1996-07-09 00:00:00.000	40	27
10253	1996-07-10 00:00:00.000	20	10
10253	1996-07-10 00:00:00.000	42	14
10253	1996-07-10 00:00:00.000	40	16

Registros a procesar

2

AÑO	TOTALACUMULADO
1998	437306
2010	10544
2007	6745
1996	222653
2008	8246
1997	651427
2011	3242
2009	81

Aplicando
GROUP BY

3

AÑO	TOTALACUMULADO
2008	8246
2009	81
2010	10544
2011	3242

Aplicando
HAVING

```
SELECT YEAR(P.FECHAPEDIDO) AS AÑO,SUM(D.CANTIDAD*D.PRECIOUNIDAD) AS TOTALACUMULADO
FROM VENTAS.PEDIDOSCABE P JOIN VENTAS.PEDIDOSDETA D ON P.IDPEDIDO=D.IDPEDIDO
GROUP BY YEAR(P.FECHAPEDIDO)
HAVING YEAR(P.FECHAPEDIDO)>=2008
GO
```

Ejemplo 02

Utilizando la tabla **DEPARTAMENTOS**

	COD_EDIF	COD_DEP	AREA_TOTAL_DEP	AREA_CONSTRUIDA_DEP	NUM_AMB_DEP	PISO_DEP	PRECIO_
1	EDF001	DPT001	250	200	6	1	200
2	EDF001	DPT002	180	144	5	1	250
3	EDF001	DPT003	220	176	5	1	280
4	EDF001	DPT004	190	152	4	1	250
5	EDF001	DPT005	230	184	6	2	280
6	EDF001	DPT006	200	160	5	2	250
7	EDF001	DPT007	230	184	6	2	240
8	EDF001	DPT008	200	160	5	2	230
9	EDF001	DPT009	260	208	6	3	320
10	EDF001	DPT010	220	176	5	3	290
11	EDF001	DPT011	225	180	7	3	280
12	EDF001	DPT012	240	192	8	3	285
13	EDF001	DPT013	240	192	8	4	285
14	EDF001	DPT014	240	192	8	4	285
15	EDF001	DPT015	240	192	8	4	285
16	EDF001	DPT016	240	192	8	4	285

Ejemplo 02

Mostrar la cantidad de departamentos agrupados por edificio.

```
SELECT COD_EDIF, COUNT(COD_DEP) AS 'CANTIDAD DPTOS'
FROM DEPARTAMENTOS
GROUP BY COD_EDIF
```

	COD_EDIF	COD_DEP	AREA_TOTAL_DEP	AREA_CONSTRUIDA_DEP	NUM_AMB_DEP	PISO_DEP	PRECIO_
1	EDF001	DPT001	250	200	6	1	200
2	EDF001	DPT002	180	144	5	1	250
3	EDF001	DPT003	220	176	5	1	280
4	EDF001	DPT004	190	152	4	1	250
5	EDF001	DPT005	230	184	6	2	280
6	EDF001	DPT006	200	160	5	2	250
7	EDF001	DPT007	230	184	6	2	240
8	EDF001	DPT008	200	160	5	2	230
9	EDF001	DPT009	260	208	6	3	320
10	EDF001	DPT010	220	176	5	3	290
11	EDF001	DPT011	225	180	7	3	280
12	EDF001	DPT012	240	192	8	3	285
13	EDF001	DPT013	240	192	8	4	285
14	EDF001	DPT014	240	192	8	4	285
15	EDF001	DPT015	240	192	8	4	285
16	EDF001	DPT016	240	192	8	4	285

GROUP BY combina registros con valores idénticos, en la lista de campos específicos, en un único registro.

Ejemplo 02

Mostrar la cantidad de departamentos agrupados por edificio.

```
SELECT COD_EDIF, COUNT(COD_DEP) AS 'CANTIDAD DPTOS'
FROM DEPARTAMENTOS
GROUP BY COD_EDIF
```

FUNCION AGREGADA

	COD_EDIF	COD_DEP	AREA_TOTAL_DEP	AREA_CONSTRUIDA_DEP	NUM_AMB_DEP	PISO_DEP	PRECIO_
1	EDF001	DPT001	250	200	6	1	200
2	EDF001	DPT002	180	144	5	1	250
3	EDF001	DPT003	220	176	5	1	280
4	EDF001	DPT004	190	152	4	1	250
5	EDF001	DPT005	230	184	6	2	280
6	EDF001	DPT006	200	160	5	2	250
7	EDF001	DPT007	230	184	6	2	240
8	EDF001	DPT008	200	160	5	2	230
9	EDF001	DPT009	260	208	6	3	320
10	EDF001	DPT010	220	176	5	3	290
11	EDF001	DPT011	225	180	7	3	280
12	EDF001	DPT012	240	192	8	3	285
13	EDF001	DPT013	240	192	8	4	285
14	EDF001	DPT014	240	192	8	4	285
15	EDF001	DPT015	240	192	8	4	285
16	EDF001	DPT016	240	192	8	4	285

Ejemplo 02

RESULTADO

	COD_EDIF	CANTIDAD DPTOS
1	EDF001	40
2	EDF002	32
3	EDF003	24
4	EDF004	36
5	EDF005	20
6	EDF006	40
7	EDF007	24
8	EDF008	24
9	EDF009	32
10	EDF010	34
11	EDF011	20
12	EDF012	24
13	EDF013	20
14	EDF014	24
15	EDF015	36
16	EDF016	32
17	EDF017	24

Ejemplo 02

APLICANDO HAVING

Mostrar la cantidad de departamentos agrupados por edificio, pero únicamente de aquellos con más de 35 departamentos.

```
SELECT COD_EDIF, COUNT(COD_DEP) AS 'CANTIDAD DPTOS'
FROM DEPARTAMENTOS
GROUP BY COD_EDIF
HAVING COUNT(COD_DEP) >=35
```

	COD_EDIF	CANTIDAD DPTOS
1	EDF001	40
2	EDF004	36
3	EDF006	40
4	EDF015	36
5	EDF025	36

RESULTADO

	COD_EDIF	CANTIDAD DPTOS
1	EDF001	40
2	EDF002	32
3	EDF003	24
4	EDF004	36
5	EDF005	20
6	EDF006	40
7	EDF007	24
8	EDF008	24
9	EDF009	32
10	EDF010	34
11	EDF011	20
12	EDF012	24
13	EDF013	20
14	EDF014	24
15	EDF015	36
16	EDF016	32
17	EDF017	24

EJEMPLOS ADICIONALES Y OTRAS CONSIDERACIONES

Actividad

```
SELECT COD_EDIF, COD_DEP
FROM DEPARTAMENTOS
ORDER BY COD_EDIF
```

Verificar las diferencias

```
SELECT COD_EDIF, COUNT(COD_DEP)
FROM DEPARTAMENTOS
GROUP BY COD_EDIF
```

CONSIDERACIONES EN EL USO DE *GROUP BY*

- SQL genera una fila por cada grupo que se especificó. **No retorna información detallada.**
- Todas las columnas que son especificadas en la claúsula **GROUP BY** deben estar incluidas en la lista de campos de la claúsula SELECT.
- Si se incluye una claúsula **WHERE** ,SQL agrupa solamente las filas que cumplen con las condiciones de la claúsula **WHERE**

CONSIDERACIONES EN *EL USO DE HAVING*

- Usar **HAVING** solamente con la claúsula **GROUP BY** para restringir grupos.

- **HAVING** aplica condiciones sobre la claúsula **GROUP BY** de la misma manera que la claúsula **WHERE** interactúa con la claúsula **SELECT.**