

Base de Datos I

SQL - SELECT Agregación

Modelo de datos

Empleado

<u>nro</u>	nombre	cod_esp	nro_jefe	sueldo	f_ingreso
1000	Juan	1		10000	1/1/2000
1001	Pedro	2	1000	5000	1/5/2008
1002	Daniel	2	1000	2000	1/10/2009

Trabaja

<u>Cod t</u>	<u>nro_emp</u>	<u>cod_area</u>
1	1000	A1
2	1000	A2
3	1001	A1
4	1002	A2

Area

<u>cod_area</u>	descripcion
A1	Area 1
A2	Area 2

Especialidad

<u>cod_esp</u>	descripcion
1	Gerente
2	Operario

Ejercicio 1

“Indicar la cantidad de empleados de la empresa”

Empleado

<u>nro</u>	nombre	<u>cod_esp</u>	<u>nro_jefe</u>	sueldo	f_ingreso
1000	Juan	1		10000	1/1/2000
1001	Pedro	2	1000	5000	1/5/2008
1002	Daniel	2	1000	2000	1/10/2009

Funciones de agregación

- COUNT (*)
 - COUNT (<campo>)
 - COUNT (DISTINCT <campo>)
- MAX(<campo>)
- MIN(<campo>)
- SUM(<campo>)
- AVG(<campo>)

Proceso de agrupamiento

Temporal

Agrupamiento

Temporal Agrupada

- Un registro por **grupo**
- Aplicar funciones de agregación

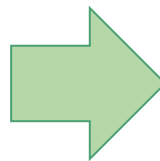
Ejercicio 1

“Indicar la cantidad de empleados de la empresa”

```
SELECT COUNT(*)  
FROM Empleado;
```

Temporal

nro	nombre	cod_esp	nro_jefe	sueldo	f_ingreso
1000	Juan	1		10000	1/1/2000
1001	Pedro	2	1000	5000	1/5/2008
1002	Daniel	2	1000	2000	1/10/2009



Temporal Agrupada

COUNT(*)
3

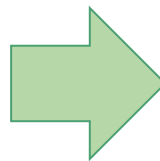
Ejercicio 2

“Indicar la cantidad de empleados y sueldo máximo de la empresa”

```
SELECT COUNT(*), MAX(sueldo)
FROM Empleado;
```

Temporal

nro	nombre	cod_esp	nro_jefe	sueldo	f_ingreso
1000	Juan	1		10000	1/1/2000
1001	Pedro	2	1000	5000	1/5/2008
1002	Daniel	2	1000	2000	1/10/2009



Temporal Agrupada

COUNT(*)	MAX(sueldo)
3	10000

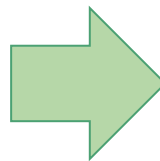
Ejercicio 3

“Cuántos empleados ganan más de \$3000?”

```
SELECT COUNT(*)  
FROM Empleado  
WHERE sueldo > 3000;
```

Temporal

nro	nombre	cod_esp	nro_jefe	sueldo	f_ingreso
1000	Juan	1		10000	1/1/2000
1001	Pedro	2	1000	5000	1/5/2008



Temporal Agrupada

COUNT(*)
2

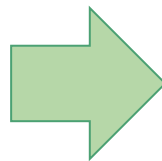
Ejercicio 4

“Indicar el sueldo mínimo de los empleados por cada código de especialidad”

```
SELECT cod_esp, MIN(sueldo)
FROM Empleado;
```

Temporal

nro	nombre	cod_esp	nro_jefe	sueldo	f_ingreso
1000	Juan	1		10000	1/1/2000
1001	Pedro	2	1000	5000	1/5/2008
1002	Daniel	2	1000	2000	1/10/2009



Temporal Agrupada

cod_esp	MIN(sueldo)
?	2000

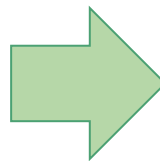
Ejercicio 4

“Indicar el sueldo mínimo de los empleados por cada código de especialidad”

FROM Empleado
GROUP BY cod_esp;

Temporal

nro	nombre	cod_esp	nro_jefe	sueldo	f_ingreso
1000	Juan	1		10000	1/1/2000
1001	Pedro	2	1000	5000	1/5/2008
1002	Daniel	2	1000	2000	1/10/2009



Temporal Agrupada

cod_esp	
1	
2	

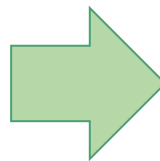
Ejercicio 4

“Indicar el sueldo mínimo de los empleados por cada código de especialidad”

```
SELECT cod_esp, MIN(sueldo)
FROM Empleado
GROUP BY cod_esp;
```

Temporal

nro	nombre	cod_esp	nro_jefe	sueldo	f_ingreso
1000	Juan	1		10000	1/1/2000
1001	Pedro	2	1000	5000	1/5/2008
1002	Daniel	2	1000	2000	1/10/2009



Temporal Agrupada

cod_esp	MIN(sueldo)
1	10000
2	2000

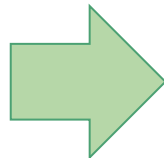
Ejercicio 5

“Indicar el sueldo mínimo de los empleados por cada código de especialidad, sólo para aquellas especialidades cuyo mínimo sea mayor a 3000”

```
SELECT cod_esp, MIN(sueldo)
FROM Empleado
WHERE MIN(sueldo) > 3000
GROUP BY cod_esp;
```

Temporal

nro	nombre	cod_esp	nro_jefe	sueldo	f_ingreso
1000	Juan	1		10000	1/1/2000
1001	Pedro	2	1000	5000	1/5/2008
1002	Daniel	2	1000	2000	1/10/2009



Temporal Agrupada

cod_esp	MIN(sueldo)
1	10000
2	2000

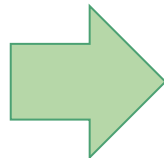
Ejercicio 5

“Indicar el sueldo mínimo de los empleados por cada código de especialidad, sólo para aquellas especialidades cuyo mínimo sea mayor a 3000”

```
SELECT cod_esp, MIN(sueldo)
FROM Empleado
GROUP BY cod_esp
HAVING MIN(sueldo) > 3000;
```

Temporal

nro	nombre	cod_esp	nro_jefe	sueldo	f_ingreso
1000	Juan	1		10000	1/1/2000
1001	Pedro	2	1000	5000	1/5/2008
1002	Daniel	2	1000	2000	1/10/2009



Temporal Agrupada

cod_esp	MIN(sueldo)
1	10000
2	2000

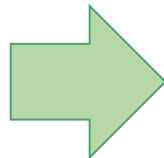
Ejercicio 5

“Indicar el sueldo mínimo de los empleados por cada código de especialidad, sólo para aquellas especialidades cuyo mínimo sea mayor a 3000”

```
SELECT cod_esp, MIN(sueldo)
FROM Empleado
GROUP BY cod_esp
HAVING MIN(sueldo) > 3000;
```

Temporal

nro	nombre	cod_esp	nro_jefe	sueldo	f_ingreso
1000	Juan	1		10000	1/1/2000
1001	Pedro	2	1000	5000	1/5/2008
1002	Daniel	2	1000	2000	1/10/2009



Temporal Agrupada

cod_esp	MIN(sueldo)
1	10000

Ejercicio 6

“Indicar el sueldo mínimo de los empleados por cada código de especialidad, sólo para aquellas especialidades con más de 5 empleados”

```
SELECT cod_esp, MIN(sueldo)
FROM Empleado
GROUP BY cod_esp
HAVING COUNT(*) > 5;
```

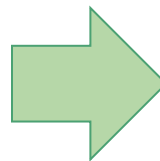
<u>nro</u>	nombre	cod_esp	nro_jefe	sueldo	f_ingreso
1000	Juan	1		10000	1/1/2000
1001	Pedro	2	1000	5000	1/5/2008
1002	Daniel	2	1000	2000	1/10/2009

Ejercicio 7

“Indicar cantidad de empleados por cada descripción de especialidad”

```
SELECT ES.descripcion, COUNT(*)  
FROM Empleado EM JOIN  
    Especialidad ES ON EM.cod_esp = ES.cod_esp  
GROUP BY ES.descripcion;
```

Empleado			Temporal		Especialidad		
nro	nombre	cod_esp	nro_jefe	suelo	f_ingreso	cod_esp	descripcion
1000	Juan	1		10000	1/1/2000	1	Gerente
1001	Pedro	2	1000	5000	1/5/2008	2	Operario
1002	Daniel	2	1000	2000	1/10/2009	2	Operario



Temporal Agrupada	
descripcion	COUNT(*)
Gerente	1
Operario	2

Ejercicio 7

Problemas:

- Que sucede si dos o más especialidades tienen la misma descripción?

Ejercicio 7

“Indicar cantidad de empleados por cada descripción de especialidad”

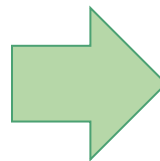
```
SELECT ES.cod_esp, ES.descripcion, COUNT(*)
```

```
FROM Empleado EM JOIN
```

```
    Especialidad ES ON EM.cod_esp = ES.cod_esp
```

```
GROUP BY ES.cod_esp, ES.descripcion;
```

Empleado			Temporal		Especialidad		
nro	nombre	cod_esp	nro_jefe	suelo	f_ingreso	cod_esp	descripcion
1000	Juan	1		10000	1/1/2000	1	Gerente
1001	Pedro	2	1000	5000	1/5/2008	2	Operario
1002	Daniel	2	1000	2000	1/10/2009	2	Operario



Temporal Agrupada	
cod_esp	COUNT(*)
1	1
2	2

Ejercicio 7

“Indicar cantidad de empleados por cada descripción de especialidad”

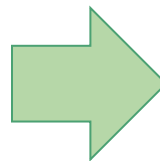
```
SELECT ES.cod_esp, ES.descripcion, COUNT(*)
```

```
FROM Empleado EM JOIN
```

```
    Especialidad ES ON EM.cod_esp = ES.cod_esp
```

```
GROUP BY ES.cod_esp, ES.descripcion;
```

Empleado			Temporal			Especialidad	
nro	nombre	cod_esp	nro_jefe	suelo	f_ingreso	cod_esp	descripcion
1000	Juan	1		10000	1/1/2000	1	Gerente
1001	Pedro	2	1000	5000	1/5/2008	2	Operario
1002	Daniel	2	1000	2000	1/10/2009	2	Operario



Temporal Agrupada		
cod_esp	descripcion	COUNT(*)
1	Gerente	1
2	Operario	2

Ejercicio 7

Solución:

- Normalmente agruparemos por campos que sean claves
- Sólo podremos devolver en la cláusula `SELECT` aquellos campos por los cuales hemos agrupado, o funciones de agregación

Ejercicio 8

“Indicar cantidad de empleados a cargo de cada jefe, mostrando su nombre”

```
SELECT J.nro, J.nombre, COUNT(*)  
FROM Empleado EM JOIN  
      Empleado J ON EM.nro_jefe = J.nro  
GROUP BY J.nro, J.nombre;
```

<u>nro</u>	nombre	cod_esp	nro_jefe	sueldo	f_ingreso
1000	Juan	1		10000	1/1/2000
1001	Pedro	2	1000	5000	1/5/2008
1002	Daniel	2	1000	2000	1/10/2009

Ejercicio 9

“Cuántos jefes hay?”

Empleado

nro	nombre	cod_esp	nro_jefe	sueldo	f_ingreso
1000	Juan	1		10000	1/1/2000
1001	Pedro	2	1000	5000	1/5/2008
1002	Daniel	2	1000	2000	1/10/2009
1003	Alejandro	2	1001	1500	1/1/2010
1004	Rafael	2	1003	1500	1/1/2010
1005	Leandro	2	1001	1500	1/1/2010
1006	Marcos	2	1001	1500	1/1/2010

Variantes de COUNT

- COUNT (*): cantidad de registros del grupo
- COUNT (<campo>): cantidad de registros del grupo, donde el campo indicado no sea NULL
- COUNT (DISTINCT <campo>): cantidad de registros **únicos** del grupo, donde el campo indicado no sea NULL (no cuenta repetidos)

Ejercicio 10

“Cuántos jefes hay?”

```
SELECT COUNT(DISTINCT nro_jefe)  
FROM Empleado;
```