# **BASES DE DATOS - 1º DAM**

# **UNIDAD DIDÁCTICA 5. SQL: CONSULTAS AVANZADAS**

## 

## **BOLETÍN DE EJERCICIOS 1**

| **Resultado de Aprendizaje 5**: Realiza consultas avanzadas sobre una base de datos, empleando agrupaciones y subconsultas. |
| --- |
| **Criterio de evaluación** |
| RA05.a Se han realizado consultas que realizan agrupaciones, diferenciando los escenarios de uso de WHERE y HAVING. |
| RA05.b Se han realizado subconsultas: escalares, simples, que devuelven más de una fila, correlacionadas o en las cláusulas FROM y JOIN. |
| RA05.c Se han realizado consultas con el operador WITH. |

## **EJERCICIO 1**

Las consultas se deben ejecutar sobre la base de datos HR, disponible en el aula virtual.

**1-** Seleccionar el salario máximo de cada departamento, mostrando el nombre del departamento y dicha cantidad.

**SELECT COALESCE**(d.department\_name, 'Sin departamento'),

**MAX**(e.salary) **AS** "maximo\_salario"

**FROM** employees e **LEFT JOIN** departments d **USING** (department\_id)

**GROUP BY** d.department\_name

**ORDER BY** maximo\_salario;

**2-** Seleccionar el salario mínimo de cada departamento, para aquellos departamentos cuyo salario mínimo sea menor que 5000$.

**SELECT COALESCE**(d.department\_name, 'Sin departamento'),

**MIN**(e.salary) **AS** "minimo\_salario"

**FROM** employees e **LEFT JOIN** departments d **USING** (department\_id)

**WHERE** e.salary < 5000

**GROUP BY** d.department\_name

**ORDER BY** minimo\_salario;

**3-** Seleccionar el número de empleados que trabajan en cada oficina, mostrando el STREET\_ADDRESS y dicho número; hay que ordenar la salida de mayor a menor.

**SELECT** l.street\_address, d.department\_name, **COUNT**(\*) **AS** "num\_empleados"

**FROM** employees e **JOIN** departments d **USING** (department\_id)

**JOIN** locations l **USING** (location\_id)

**GROUP BY** l.street\_address, d.department\_name

**ORDER BY** num\_empleados **DESC**;

**No es la mejor manera de hacerlo, pero mal no está. Está mejor hecho el ejercicio 4, recomendable verlo.**

**4-** Modificar la consulta anterior para que muestre las localizaciones que no tienen ningún empleado.

**SELECT** l.street\_address, d.department\_name, **COUNT**(\*) **AS** "num\_empleados"

**FROM** employees e **JOIN** departments d **USING** (department\_id)

**JOIN** locations l **USING** (location\_id)

**GROUP BY** l.street\_address, d.department\_name

**HAVING COUNT**(\*) = 0

**ORDER BY** num\_empleados **DESC**;

SELECT **COALESCE(l.street\_address, 'Sin ubicación')**, **COALESCE(d.department\_name, 'Sin departamento')**,

**COUNT(e.employee\_id)** AS "num\_empleados"

FROM employees e **FULL JOIN** departments d USING (department\_id)

**FULL JOIN** locations l USING (location\_id)

GROUP BY l.street\_address, d.department\_name

ORDER BY num\_empleados DESC;

**Se utiliza un FULL JOIN para que las tablas tengan igualdad de preferencia entre ellas.**

**Se utiliza COALESCE en esas 2 columnas del SELECT porque ambas pueden tener un “null” en la consulta resuelta.**

**Se utiliza COUNT(e.employee\_id) en lugar de COUNT(\*) porque de esta forma se calculan además los empleados con algún dato nulo. Si se usa COUNT(\*) eso se evade.**

**No hace falta utilizar el HAVING en esta consulta.**

**5-** Seleccionar el salario que es cobrado a la vez por más personas. Mostrar dicho salario y el número de personas.

**SELECT** salary, COUNT(\*) **AS** "num\_empleados"

**FROM** employees

**GROUP BY** salary

**HAVING COUNT**(\*) > 1

**ORDER BY** salary **DESC**;

**Tampoco es la mejor manera de resolver este ejercicio pero mal no está. Se debería de haber hecho con una subconsulta (Todavía no las hemos dado).**

**6-** Seleccionar el número de empleados que empezaron a trabajar cada año, ordenando la salida por año.

**SELECT** TO\_CHAR(hire\_date, 'YYYY'), **COUNT**(\*) **AS** "num\_empleados"

**FROM** employees

**GROUP BY** TO\_CHAR(hire\_date, 'YYYY')

**ORDER BY** TO\_CHAR(hire\_date, 'YYYY');