

BB.DD

TAREA 2.6



Álvaro Gracia Aledo

20/01/2024

1º DAM

1. Partiendo de la tabla EMPLE, visualiza por cada oficio de los empleados del departamento 'VENTAS' la suma de salarios

```
SELECT SUM(e.salario) AS "SALARIO MAX", e.oficio
FROM emple e
JOIN depart d ON e.dept_no = d.dept_no
WHERE d.dnombre = 'VENTAS'
GROUP BY e.oficio;
```

2. Desde la tabla EMPLE, visualiza el departamento que tenga más empleados cuyo oficio sea EMPLEADO

```
Select dept_no, dnombre
from depart
where dept_no = (
    select dept_no
    from emple
    where oficio = 'EMPLEADO'
    group by dept_no
    having count(*) = (
        select max(count(*))
        from emple
        where oficio = 'EMPLEADO'
        group by dept_no
    )
);
```

3. Selecciona aquellos apellidos de la tabla EMPLE cuyo salario sea mayor a la media del salario en su departamento. Muestra el departamento, el apellido y el salario.

```
SELECT d.dnombre AS departamento, e.apellido, e.salario
FROM emple e
JOIN depart d ON e.dept_no = d.dept_no
WHERE e.salario > (SELECT AVG(salario) FROM emple WHERE dept_no = e.dept_no)
```

4. Obtén la especialidad con menos profesores.

```
SELECT COUNT(E.ESPECIALIDAD) AS "cantidad profesores", E.ESPECIALIDAD
FROM PROFESORES E
GROUP BY E.ESPECIALIDAD
HAVING COUNT(E.ESPECIALIDAD) = (SELECT MIN(COUNT(E.ESPECIALIDAD))
                                FROM PROFESORES E
                                GROUP BY E.ESPECIALIDAD)
ORDER BY COUNT(E.ESPECIALIDAD) ASC;
```

5. Obtén el banco con más sucursales. Los datos a obtener son “Nombre del banco” y “N° de sucursales”.

```
SELECT B.NOMBRE_BANC AS "NOMBRE DEL BANCO", COUNT(S.COD_BANCO) AS
"N DE SUCURSALES"
FROM BANCOS B
JOIN SUCURSALES S ON B.COD_BANCO = S.COD_BANCO
GROUP BY B.NOMBRE_BANC
HAVING COUNT(S.COD_BANCO) = (SELECT MAX(COUNT(S.COD_BANCO))
                              FROM SUCURSALES S
                              GROUP BY S.COD_BANCO)
ORDER BY COUNT(S.COD_BANCO) DESC;
```

6. Obtén 1 fila por cada banco de ‘GUADALAJARA’ de su saldo actual. Los campos a mostrar son “Nombre Banco”, “Saldo Debe”, “Saldo Haber”.

```
SELECT B.NOMBRE_BANC AS "NOMBRE BANCO", SUM(C.SALDO_DEBE) AS
"SALDO DEBE", SUM(C.SALDO_HABER) AS "SALDO HABER"
FROM BANCOS B
INNER JOIN CUENTAS C ON B.COD_BANCO = C.COD_BANCO
WHERE B.POBLACION = 'GUADALAJARA'
GROUP BY B.NOMBRE_BANC;
```

7. Datos de la cuenta o cuentas con más movimientos mostrando el “Nombre Cta” y “N° Movimientos”.

```
SELECT C.NOMBRE_CTA AS "NOMBRE DE CTA", COUNT(M.COD_BANCO) AS "N
MOVIMIENTOS"
FROM CUENTAS C
JOIN MOVIMIENTOS M ON C.COD_BANCO = M.COD_BANCO
GROUP BY C.NOMBRE_CTA
HAVING COUNT(*) = (SELECT MAX(COUNT(M.COD_BANCO))
                    FROM MOVIMIENTOS M
                    GROUP BY M.COD_BANCO)
ORDER BY COUNT(M.TIPO_MOV) DESC;
```

8. El nombre de la sucursal que haya tenido más suma de reintegros//

```
SELECT S.NOMBRE_SUC AS "NOMBRE SUCURSAL", COUNT(*) AS REINTEGROS
FROM SUCURSALES S
INNER JOIN MOVIMIENTOS M ON S.COD_SUCUR = M.COD_SUCUR
WHERE M.TIPO_MOV = 'R'
GROUP BY S.NOMBRE_SUC
HAVING COUNT(*) = (SELECT MAX(COUNT(*))
                    FROM MOVIMIENTOS M
                    WHERE M.TIPO_MOV = 'R'
                    GROUP BY M.COD_SUCUR);
```

9. A partir de la tabla EMPLE y DEPART visualiza el número de departamento y el nombre de departamento que tenga más empleados cuyo oficio sea EMPLEADO.

```
SELECT E.DEPT_NO AS "N DEPARTAMENTO", D.DNOMBRE AS "NOMBRE DEPART",
COUNT(*) AS "N EMPLEADOS"
FROM EMPLE E
JOIN DEPART D ON D.DEPT_NO = E.DEPT_NO
WHERE E.OFICIO = 'EMPLEADO'
GROUP BY E.DEPT_NO, D.DNOMBRE
HAVING COUNT(*) > (SELECT MAX(COUNT(*))
                    FROM DEPART D
                    GROUP BY D.DEPT_NO)
ORDER BY COUNT(*) DESC;
```

10. Visualiza los nombres de los alumnos de la tabla ALUM que aparezcan en cualquiera de estas dos tablas: ANTIGUOS y NUEVOS. (4 registros)

```
SELECT a.nombre  
FROM alum a  
JOIN antiguos an ON a.nombre = an.nombre  
UNION  
SELECT a.nombre  
FROM alum a  
JOIN nuevos n ON a.nombre = n.nombre;
```

11. Escribe una o varias formas en que se puede poner la consulta anterior llegando al mismo resultado.

```
SELECT a.nombre  
FROM alum a  
LEFT JOIN antiguos an ON a.nombre = an.nombre  
LEFT JOIN nuevos n ON a.nombre = n.nombre  
WHERE an.nombre IS NOT NULL OR n.nombre IS NOT NULL;
```

12. Visualiza aquellos nombres de la tabla ALUM que no están en la tabla ANTIGUOS ni en la tabla NUEVOS. (3 registros)

```
SELECT a.nombre  
FROM alum a  
LEFT JOIN antiguos an ON a.nombre = an.nombre  
LEFT JOIN nuevos n ON a.nombre = n.nombre  
WHERE an.nombre IS NULL AND n.nombre IS NULL;
```

13. Realiza una consulta en la que aparezca por cada centro y en cada especialidad el número de profesores. Si el centro no tiene profesores, debe aparecer un 0 en la columna de número de profesores. Las columnas a visualizar son: nombre del centro, especialidad y número de profesores. (7 registros)

14. Obtén por cada centro el número de empleados (tabla PERSONAL). Si el centro carece de empleados, ha de aparecer un 0 como número total de empleados. (5 registros)

```
SELECT c.cod_centro, COUNT(p.dni) AS num_empleados
FROM centros c
LEFT JOIN personal p ON c.cod_centro = p.cod_centro
where p.funcion = 'profesor'
GROUP BY c.cod_centro;
```

15. Repite la consulta anterior pero sólo para los empleados que no son profesores (personal ADMINISTRATIVO y CONSERJE) (5 registros)

```
SELECT c.cod_centro, COUNT(p.dni) AS num_empleados
FROM centros c
LEFT JOIN personal p ON c.cod_centro = p.cod_centro
where p.funcion != 'profesor'
GROUP BY c.cod_centro;
```