Nombre y Apellidos:

Dada una base de datos denominada "movil", realiza las siguientes gestiones y consultas en SQL. El hecho de que la consulta realizada proporcione la misma salida no implica que la consulta sea correcta.

- 1. Modifica la base de datos de acuerdo a las siguientes instrucciones: [1,5 puntos]
 - Crear la tabla "mercado" de acuerdo al modelo entidad relación proporcionado, eligiendo el tipo y el tamaño adecuado para cada atributo.

• Cambiar el nombre del atributo 'unidades' a 'mercadounidades'.

alter table mercado rename unidades to mercadounidades;

• Modifica la tabla y añade el atributo 'anio' para almacenar en que año fueron vendidas las unidades.

alter table mercado add anio integer not null;

• Borra el atributo "anio".

alter table mercado drop anio;

• 'Oppo' y 'Fairphone' han vendido 16,5 y 190 millones de unidades respectivamente. Insertar las tuplas correspondientes en la tabla 'mercado', y elige el valor para el atributo 'id'.

insert into mercado values(1, 16500000, 5); // para Oppo insert into mercado values(2, 190000000, 2); // para Fairphone

• Presenta el nombre de los móviles que han vendido unidades o no unidades, junto con el número de unidades vendidas.

select mv.nombre, me.unidades from movil as mv left join mercado as me on (me.id movil = mv.id);



2. Presenta el nombre de aquellos componentes que contengan 'magnesio' o 'galio' en orden descendente. (join, in) [1,5 puntos]

select c.nombre
from componentematerial as cm
inner join componente as c on (c.id = cm.id_componente)
inner join material as m on (m.id = cm.id_material)
where lower(m.nombre) = 'magnesio' or lower(m.nombre) = 'galio'
order by c.nombre desc;



3. Presenta el nombre de aquellos componentes con al menos dos materiales cuya primera letra empiece por 'n'. [1,5 puntos]

select nombre, numero_materiales from (
select c.nombre, count(*) as numero_materiales from componentematerial as cm
inner join material as m on (m.id = cm.id_material)
inner join componente as c on (c.id = cm.id_componente) where lower(m.nombre) like 'n%'
group by c.nombre) as asm
where numero_materiales >= 2;

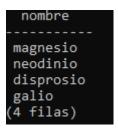
```
nombre | numero_materiales
-----
Microfono | 2
(1 fila)
```



)

4. Presenta el nombre de aquellos materiales con origen desconocido. [1.5 points]

(select nombre from material)
except
(select m.nombre
from materialorigen as mo
inner join material as m on (m.id = mo.id_material));



5. Presenta aquellos materiales que provengan de dos lugares y aquellos que provienen de 'Rusia'. (UNION) [2 puntos]

Los que vienen de dos lugares:

select id from
(select m.id, count(*) as count
from materialorigen as mo
inner join material as m on (m.id = mo.id_material)
inner join origen as o on (o.id = mo.id_origen)
group by m.id) as aux
where count = 2;

Los que vienen de Rusia:

select m2.id from materialorigen as mo2 inner join material as m2 on (m2.id = mo2.id_material) inner join origen as o2 on (o2.id = mo2.id_origen) where lower(o2.lugar) = 'rusia';

Unión de ambos:

Incluye el nombre, que se obtiene por un producto cartesiano de la unión y la tabla 'material'.

select m.id, m.nombre

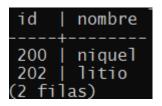
from (

(select id from (select m.id, count(*) as count from materialorigen as mo inner join material as m on (m.id = mo.id_material) inner join origen as o on (o.id = mo.id_origen) group by m.id) as aux where count = 2)



union

(select m2.id from materialorigen as mo2 inner join material as m2 on (m2.id = mo2.id_material) inner join origen as o2 on (o2.id = mo2.id_origen) where lower(o2.lugar) = 'rusia')
) as final inner join material as m using (id);



6. Presenta el nombre de todos aquellos componentes que contengan el material 'niquel'. [2 points]

select distinct nombre from componente where not exists(
(select c1.id from componentematerial as cm1 inner join componente as c1 on (c1.id = cm1.id_componente) inner join material as m1 on (m1.id = cm1.id_material)) except
(select c2.id from componentematerial as cm2 inner join componente as c2 on (c2.id = cm2.id_componente) inner join material as m2 on (m2.id = cm2.id_material) where lower(m2.nombre) = 'niquel')
);

Todos los nombres de los componentes que no existan en la lista de (Todos los componentes – los componentes que contengan niquel) =>

Todos los nombres de los componentes que no existan en la lista de (Todos los componentes que no contengan niquel) =>

Todos los componentes que contengan niquel.

nombre
Electronica
Caja
Bateria
Microfono
Tarjeta MicroSD
Plegable
(6 filas)