

/\*

Almacen (nro, nombre, responsable)

Articulo (cod\_art, descripción, precio)

Material (cod\_mat, descripción)

Proveedor (cod\_prov, nombre, domicilio, cod\_ciu, fecha\_alta)

Ciudad (cod\_ciu, nombre)

Contiene (nro, cod\_art)

Compuesto\_por (cod\_art, cod\_mat)

Provisto\_por (cod\_mat, cod\_prov)

\*/

-- 1 Listar los números de artículos cuyo precio se encuentre entre \$100 y \$1000 y su descripción comience con la letra A

```
SELECT cod_art , descripcion
      FROM articulo
      WHERE precio BETWEEN 100
            AND 1000 AND descripcion LIKE 'A%';
```

-- 2 Listar todos los datos de todos los proveedores

```
SELECT *
FROM proveedor;
```

-- 3 Listar la descripción de los materiales de código 1, 3, 6, 9 y 18.

```
SELECT descripcion
      FROM material
      WHERE cod_mat IN (1,3,6,9,18);
```

-- 4 Listar código y nombre de proveedores de la calle Suipacha, que hayan sido dados de alta en el año 2001.

```
SELECT cod_prov, nombre
      FROM proveedor
      WHERE domicilio
            LIKE '%Suipacha%'
            AND fecha_alta < '2002-01-01' AND fecha_alta >= '2001-01-01';
```

-- 5Listar nombre de todos los proveedores y de su ciudad. (Forma Explicita) (Usa 2 tablas)

```
SELECT PRO.nombre, CIU.nombre  
  
FROM proveedor PRO  
  
LEFT JOIN ciudad CIU ON PRO.cod_ciu=CIU.cod_ciu;
```

-- 6Listar los nombres de los proveedores de la ciudad de La Plata. (Forma implicita) (Usa 2 tablas)

```
SELECT pro.nombre, ciu.nombre  
  
FROM proveedor pro, ciudad ciu  
  
WHERE pro.cod_ciu= ciu.cod_ciu  
  
AND ciu.nombre='La Plata';
```

-- 7Listar los números de almacenes que almacenan el artículo de descripción A (Usa 3 tablas)

```
SELECT AL.nro  
  
FROM almacen AL  
  
JOIN contiene C ON AL.nro=C.nro  
  
JOIN articulo ART ON ART.cod_art=C.cod_art  
  
WHERE ART.descripcion='A';
```

-- 8Listar los materiales (código y descripción) provistos por proveedores de la ciudad de Rosario

```
SELECT M.cod_mat, M.descripcion  
  
FROM material M  
  
JOIN provisto_por P ON M.cod_mat=P.cod_mat  
  
JOIN proveedor Pro ON Pro.cod_prov=P.cod_prov  
  
JOIN ciudad CIU ON Pro.cod_ciu=CIU.cod_ciu  
  
WHERE CIU.nombre='Rosario';
```

-- 9 Listar los nombres de los proveedores que proveen materiales para artículos ubicados en almacenes que Martín Gómez tiene a su cargo.

```
SELECT DISTINCT PRO.nombre
FROM proveedor PRO
  JOIN provisto_por P ON PRO.cod_prov=P.cod_prov
  JOIN material M ON P.cod_mat=M.cod_mat
  JOIN compuesto_por COM ON M.cod_mat=COM.cod_mat
  JOIN articulo ART ON ART.cod_art=COM.cod_art
  JOIN contiene C ON ART.cod_art=C.cod_art
  JOIN almacen ALM ON ALM.nro=C.nro
WHERE ALM.responsable LIKE '%Martin%Gomez%';
```

-- 10 Mover los artículos almacenados en el almacén de número 10 al de número 20.

```
UPDATE contiene
SET nro=20
WHERE nro=10;
```

-- 11 Eliminar a los proveedores que contengan la palabra ABC en su nombre

```
DELETE FROM proveedor
WHERE nombre LIKE '%ABC%';
```

-- 12 Indicar la cantidad de proveedores que comienzan con la letra F

```
SELECT count(*) as 'Nombres con F'
FROM proveedor
WHERE nombre LIKE 'F%';
```

-- 13 Listar el promedio de precios de los artículos por cada almacén (nombre)

```
SELECT ALM.nombre as 'Almacenes', AVG(ART.precio) as 'PrecioPromedio'
FROM almacen ALM
  JOIN contiene C ON ALM.nro=C.nro
  JOIN articulo ART ON ART.cod_art=C.cod_art
GROUP BY ALM.nro, ALM.nombre;
```

-- 14Listar la descripción de artículos compuestos por al menos 2 materiales

```
SELECT A.descripcion
      FROM articulo A
      JOIN compuesto_por C ON A.cod_art=C.cod_art
      GROUP BY A.cod_art, A.descripcion
      HAVING COUNT(*)>=2;
```

-- 15Listar cantidad de materiales que provee cada proveedor (código, nombre y domicilio)

```
SELECT p.cod_prov, p.nombre, p.domicilio,
COUNT(pp.cod_mat) AS 'cantidad de materiales'
      FROM proveedor p
      LEFT JOIN provisto_por pp ON p.cod_prov = pp.cod_prov
      GROUP BY p.cod_prov, p.nombre, p.domicilio;
```

-- 16Cuál es el precio máximo de los artículos que proveen los proveedores de la ciudad de Zárate

```
SELECT max(a.precio) FROM proveedor p
      JOIN provisto_por pp ON p.cod_prov = pp.cod_prov
      JOIN compuesto_por cp ON pp.cod_mat = cp.cod_mat
      JOIN articulo a ON cp.cod_art = a.cod_art
      JOIN ciudad c ON c.cod_ciu = p.cod_ciu
      WHERE c.nombre = 'Zarate';
```

-- 17Listar los nombres de aquellos proveedores que no proveen ningún material

```
SELECT P.nombre
      FROM proveedor P
      WHERE NOT EXISTS (SELECT 1
                        FROM Provisto_por pp
                        WHERE pp.cod_prov=P.cod_prov);
```

-- 18) Listar los códigos de los materiales que provea el proveedor 10 y no los provea el proveedor 15.

```
SELECT pp.cod_mat
FROM provisto_por pp
WHERE pp.cod_prov=10
AND pp.cod_mat NOT IN(SELECT pp2.cod_mat
FROM provisto_por pp2
WHERE pp2.cod_prov=15);
```

-- 19) Listar número y nombre de almacenes que contienen los artículos de descripción A y los de descripción B (ambos).

```
SELECT a.nro, a.nombre
FROM almacen a
WHERE a.nro IN (SELECT c.nro
FROM contiene c
JOIN articulo a ON c.cod_art=a.cod_art
WHERE a.descripcion='A')
AND a.nro IN (SELECT c.nro
FROM contiene c
JOIN articulo a ON c.cod_art=a.cod_art
WHERE a.descripcion='B');
```

-- 20) Listar la descripción de artículos compuestos por todos los materiales.

```
SELECT a.descripcion
FROM articulo a
JOIN compuesto_por cp ON a.cod_art=cp.cod_art
GROUP BY a.cod_art, a.descripcion
HAVING COUNT (cp.cod_mat)=(SELECT(cod_mat)
FROM material);
```

-- 21 Hallar los códigos y nombres de los proveedores que proveen al menos un material que se usa en algún artículo cuyo precio es mayor a \$100.

```
SELECT p.cod_prov, p.nombre
      FROM proveedor p
      WHERE EXISTS (SELECT 1
                    FROM provisto_por pp
                    WHERE p.cod_prov=pp.cod_prov
                    AND EXISTS (SELECT 1
                                FROM compuesto_por cp
                                JOIN articulo a ON cp.cod_art=a.cod_art
                                WHERE a.precio>100 AND pp.cod_mat=cp.cod_mat));
```

-- 22 Listar la descripción de los artículos de mayor precio.

```
SELECT a.descripcion
      FROM articulo
      WHERE precio=(SELECT MAX(precio)
                    FROM articulo);
```

-- 23) Listar los nombres de proveedores de Capital Federal que sean únicos proveedores de algún material.

```
SELECT p.nombre
      FROM proveedor p JOIN ciudad c ON p.cod_ciu= c.cod_ciu
      WHERE c.nombre='Capital Federal'
      AND p.cod_prov IN (SELECT MIN(pp.cod_prov)
                        FROM provisto_por pp
                        group by pp.cod_mat
                        having count(*)=1);
```

-- 24Listar los nombres de almacenes que almacenan la mayor cantidad de artículos.

```
SELECT a.nombre
      from almacen a join contiene c on a.nro=c.nro
      group by a.nro, a.nombre
      having count(*)=(SELECT MAX(total)
                        from(select count(*)total
                        from contiene c
                        group by c.nro)
                        alias);
```

-- 25Listar la ciudades donde existan proveedores que proveen todos los materiales

```
SELECT c.nombre
      FROM ciudad c join proveedor p on c.cod_ciu= p.cod_ciu
      group by c.nombre
      having count(*)=(SELECT MAX(total)
                        from(select count(*) total
                        from provisto_por pp
                        group by pp.cod_mat)
                        alias);
```

-- 26Listar los números de almacenes que tienen todos los artículos que incluyen el material con código 123.

```
SELECT a.nro
      from almacenes a join contiene c on a.cod_art=c.cod_art
      group by numeroDeAlmacen
      having count(*)=(SELECT art.cod_art
                        from articulo art join compuesto_por cp on art.cod_art=cp.cod_art
                        where cp.cod_mat=123);
```

## EJERCICIO 2

/\*Vuelo (nro\_vuelo, desde, hasta, fecha, nro\_avion)

Avion(nro\_avion, tipo\_avion, modelo)

Pasajero(nro\_vuelo, documento, nombre)\*/

-- 1Hallar los números de vuelo desde el origen A hasta el destino F

```
select v.nro_vuelo
      from vuelo v
      where v.desde='A' and v.hasta='F';
```

-- 2Hallar los nombres de pasajeros y números de vuelo para aquellos pasajeros que viajan desde A a D.

```
select p.nombre, p.nro_vuelo
      from pasajero p join vuelo v on p.nro_vuelo=v.nro_vuelo
      where v.desde='A' and v.hasta='D';
```

-- 3Hallar los tipos de avión para vuelos que parten desde C.

```
select a.tipo_avion
      from avion join vuelo v on a.nro_avion=v.nro_avion
      where v.desde='C';
```

-- 4Listar los distintos tipo y nro. de avión que tienen a H como destino

```
select a.tipo_avion, a.nro_avion
      from avion a
      where a.hasta='H';
```

-- 5Mostrar por cada Avión (número y modelo) la cantidad de vuelos en que se encuentra registrado.

```
select a.nro_avion, a.modelo, count(*)
      from avion a join vuelo v on a.nro_vuelo=v.nro_vuelo
      group by a.nro, a.modelo;
```

-- 6Cuántos pasajeros diferentes han volado en un avión de modelo 'B-777'

```
select distinct count(*)
      from pasajero p
      where nro_vuelo=(select nro_vuelo v
                        from vuelo join avion on a.nro_avion=v.nro_avion
                        where a.modelo='B-777');
```



-- 7Listar la cantidad promedio de pasajeros transportados por los aviones de la compañía, por tipo de avión.

```
SELECT vcp.tipo_avion, AVG(cant_pasajeros) FROM
    FROM (SELECT avion.tipo_avion, COUNT(*) cant_pasajeros
        FROM avion JOIN vuelo ON avion.nro_avion = vuelo.nro_avion
        JOIN pasajero ON vuelo.nro_vuelo = pasajero.nro_vuelo
        GROUP BY pasajero.nro_vuelo, avion.tipo_avion) vcp
    GROUP BY vcp.tipo_avion;
```

-- 8Hallar los tipos de avión que no son utilizados en algún vuelo que pase por B.

```
select a.tipo_avion
    from avion a join vuelo v on a.nro_avion=v.nro_avion
    where a.nro_avion NOT IN (SELECT nro_avion
        FROM vuelo
        WHERE (desde = 'B'
            OR hasta = 'B')
            AND v.nro_avion = a.nro_avion)
    GROUP BY a.nro_avion;
```

### EJERCICIO 3

/\*Producto (id\_producto, descripcion, estado, id\_proveedor)

Proveedor (id\_proveedor, nombre, cuit)

Cliente (id\_cliente, nombre)

Vendedor (id\_empleado, nombre, apellido, dni)

Venta (nro\_factura, id\_cliente, fecha, id\_vendedor)

Detalle\_Venta (nro\_factura, nro\_detalle, id\_producto, cantidad, precio\_unitario)\*/

-- 1. Listar la cantidad de productos que tiene la empresa.

```
SELECT COUNT(producto.id_producto) AS cant_productos
FROM producto;
```

-- 2. Listar la descripción de productos en estado 'en stock' que tiene la empresa.

```
SELECT producto.descripcion  
      FROM producto  
     WHERE estado = 'activo';
```

-- 3. Listar los productos que nunca fueron vendidos.

```
SELECT producto.id_producto  
      FROM producto  
     WHERE producto.id_producto NOT IN (SELECT detalle_venta.id_producto  
                                         FROM detalle_venta);
```

-- 4. Listar la cantidad total de unidades que fueron vendidas de cada producto (descripción).

```
SELECT producto.descripcion, SUM(detalle_venta.cantidad) AS cant_vendida  
      FROM producto JOIN detalle_venta ON producto.id_producto = detalle_venta.id_producto  
     GROUP BY producto.descripcion;
```

-- 5. Listar el nombre de cada vendedor y la cantidad de ventas realizadas en el año 2015.

```
SELECT vendedor.nombre, COUNT(venta.nro_factura) AS cant_ventas  
      FROM vendedor JOIN venta ON vendedor.id_vendedor = venta.id_vendedor  
     WHERE venta.fecha LIKE '2015%'  
     GROUP BY vendedor.nombre;
```

-- 6. Listar el monto total vendido por cada cliente (nombre)

```
SELECT cliente.nombre, SUM(detalle_venta.precio_unitario * detalle_venta.cantidad) AS  
cant_comprada  
      FROM cliente JOIN venta ON cliente.id_cliente = venta.id_cliente  
     JOIN detalle_venta ON venta.nro_factura = detalle_venta.nro_factura  
     GROUP BY cliente.nombre;
```

-- 7. Listar la descripción de aquellos productos en estado 'sin stock' que se hayan vendido en el mes de Enero de 2015

```
SELECT producto.descripcion
      FROM producto JOIN detalle_venta
      ON producto.id_producto = detalle_venta.id_producto
      JOIN venta ON detalle_venta.nro_factura = venta.nro_factura
      WHERE venta.fecha LIKE '2015-01%'
      AND producto.estado = 'Sin stock';
```

#### EJERCICIO 4

/\*Persona (nro\_persona, nombre, calle, ciudad, nro\_supervisor)

Empresa(nro\_empresa, razon\_social, calle, ciudad)

Trabaja (nro\_persona, nro\_empresa, salario, fecha\_ingreso)

Nota: las claves primarias se marcaron en negrita, y las claves foráneas subrayadas\*/

-- 1. Listar el nombre y ciudad de todas las personas que trabajan en la empresa "Banelco".

```
SELECT persona.nombre, persona.ciudad
      FROM persona JOIN trabaja ON persona.nro_persona = trabaja.nro_persona
      JOIN empresa ON trabaja.nro_empresa = empresa.nro_empresa
      WHERE empresa.razon_social = 'Banelco';
```

-- 2. Listar el nombre, calle y ciudad de todas las personas que trabajan para la empresa "Paulinas" y ganan más de \$1500.

```
SELECT persona.nombre, persona.calle, persona.ciudad
      FROM persona JOIN trabaja ON persona.nro_persona = trabaja.nro_persona
      JOIN empresa ON trabaja.nro_empresa = empresa.nro_empresa
      WHERE empresa.razon_social = 'Paulinas'
      AND trabaja.salario > 1500;
```

-- 3. Listar el nombre de personas que viven en la misma ciudad en la que se halla la empresa en donde trabajan.

```
SELECT persona.nombre
      FROM persona JOIN trabaja ON persona.nro_persona = trabaja.nro_persona
      JOIN empresa ON trabaja.nro_empresa = empresa.nro_empresa
```

```
WHERE persona.ciudad = empresa.ciudad
```

```
GROUP BY persona.nombre;
```

-- 4. Listar número y nombre de todas las personas que viven en la misma ciudad y en la misma calle que su supervisor.

```
SELECT pp.nro_persona, pp.nombre
```

```
FROM persona p JOIN persona pp ON p.nro_persona = pp.nro_supervisor
```

```
WHERE p.ciudad = pp.ciudad
```

```
AND p.calle = pp.calle
```

```
AND p.nro_persona != pp.nro_persona;
```

-- 5. Listar el nombre y ciudad de todas las personas que ganan más que cualquier empleado de la empresa "Tecnosur".

```
SELECT persona.nombre, persona.ciudad, trabaja.salario
```

```
FROM persona JOIN trabaja ON persona.nro_persona = trabaja.nro_persona
```

```
JOIN empresa ON trabaja.nro_empresa = empresa.nro_empresa
```

```
WHERE trabaja.salario > (SELECT MAX(trabaja.salario)
```

```
FROM trabaja JOIN empresa
```

```
ON trabaja.nro_empresa = empresa.nro_empresa
```

```
WHERE empresa.razon_social = 'Tecnosur');
```

-- 6. Listar las ciudades en las que todos los trabajadores que vienen en ellas ganan más de \$1000.

```
SELECT persona.ciudad, COUNT(*) cant_empleados
```

```
FROM persona JOIN trabaja ON persona.nro_persona = trabaja.nro_persona
```

```
GROUP BY persona.ciudad;
```

```
SELECT persona.ciudad, COUNT(*) cant_salarios
```

```
FROM trabaja JOIN persona ON trabaja.nro_persona = persona.nro_persona
```

```
WHERE trabaja.salario > 1000
```

```
GROUP BY .ciudad;
```

-- 7. Listar el nombre de los empleados que hayan ingresado en mas de 4 Empresas en el periodo 01-01-2000 y 31-03-2004

```
SELECT persona.nombre, COUNT(*) cant_ingresos
```

```
FROM persona JOIN trabaja ON persona.nro_persona = trabaja.nro_persona
```

```
JOIN empresa ON trabaja.nro_empresa = empresa.nro_empresa
```

```

WHERE persona.nro_persona IN (SELECT trabaja.nro_persona
FROM trabaja
WHERE trabaja.fecha_ingreso
BETWEEN '2000-01-01' AND '2004-03-31'
GROUP BY trabaja.nro_persona)
GROUP BY persona.nro_persona;

```

## EJERCICIO 5

/\*Auto (patente, modelo, año)

Chofer (nro\_chofer, nombre, fecha\_ingreso, telefono)

Viaje (nro\_chofer, nro\_cliente, patente, fecha, km\_totales, tiempo\_espera)

Cliente (nro\_cliente, nombre, domicilio, localidad)\*/

-- 1. Cuál es el tiempo de espera promedio de los viajes del año 2005?

```

SELECT AVG(viaje.tiempo_espera) as prom_tiempo_espera
FROM viaje
WHERE viaje.fecha LIKE '2021%';

```

-- 2. Listar el nombre de los clientes que hayan viajado en todos los autos.

```

SELECT COUNT(auto.patente)
FROM auto;

SELECT cliente.nombre
FROM cliente JOIN viaje ON cliente.nro_cliente = viaje.nro_cliente
WHERE viaje.patente IN (SELECT auto.patente
FROM auto
GROUP BY auto.patente);

```

-- 3. Listar nombre y teléfono de los choferes que no manejaron algún vehículo de modelo posterior al año 2010.

```
SELECT chofer.nombre, chofer.telefono
      FROM chofer JOIN viaje ON chofer.nro_chofer = viaje.nro_chofer
      JOIN auto ON viaje.patente = auto.patente
      WHERE auto.anio NOT IN (SELECT auto.anio
                              FROM auto
                              WHERE auto.anio < 2010)
      GROUP BY chofer.nro_chofer;
```

-- 4. Listar los kilómetros realizados en viajes por cada auto (patente y modelo)

```
SELECT auto.patente, auto.modelo, SUM(viaje.km_totales)
      FROM auto JOIN viaje ON auto.patente = viaje.patente
      GROUP BY auto.patente;
```

-- 5. Mostrar el el costo promedio de los viajes realizados por cada auto (patente), para viajes de clientes de la localidad de Ramos Mejía.

```
SELECT viaje.patente, AVG(viaje.tiempo_espera)
      FROM viaje JOIN cliente ON viaje.nro_cliente = cliente.nro_cliente
      WHERE cliente.localidad = 'Ramos Mejía'
      GROUP BY viaje.patente;
```

-- 6. Listar el costo total de los viajes realizados por cada chofer (número y nombre) cuyo nombre comienza con la letra A.

```
SELECT chofer.nro_chofer, chofer.nombre, SUM(viaje.km_totales)
      FROM chofer JOIN viaje ON chofer.nro_chofer = viaje.nro_chofer
      WHERE chofer.nombre LIKE 'A%'
      GROUP BY chofer.nro_chofer;
```

-- 7. Mostrar la fecha de ingreso, el nombre del chofer y nombre de cliente, que hayan realizado el viaje más largo de la historia.

```
SELECT chofer.nombre AS chofer, cliente.nombre AS cliente, chofer.fecha_ingreso  
FROM chofer JOIN viaje ON chofer.nro_chofer = viaje.nro_chofer  
JOIN cliente ON viaje.nro_cliente = cliente.nro_cliente  
WHERE viaje.km_totales = (SELECT MAX(viaje.km_totales)  
FROM viaje);
```