**Base de datos. DDL**

Crea una base de datos llamada PRODUCCION.





1. Construir una tabla con los códigos de productos suministrados ya sea por un fabricante de Londres o a una empresa de Londres.

**CREATE TABLE productos\_londres(**

**SELECT distinct cod\_prod**

**FROM entrega**

**WHERE cod\_fab IN (SELECT cod\_fab**

**FROM fabricante**

**WHERE ciudad\_fab="Londres")**

**OR cod\_emp IN (select cod\_emp**

**FROM empresa**

**WHERE ciudad\_emp = "Londres")**

**);**

1. Construir una tabla con los códigos de empresas situadas en Londres o a las que suministre productos algún fabricante de Londres.

**CREATE TABLE empresas\_londres(**

**SELECT empresa.cod\_emp**

**FROM (entrega JOIN empresa ON entrega.cod\_emp=empresa.cod\_emp)**

**JOIN fabricante ON fabricante.cod\_fab=entrega.cod\_fab**

**WHERE ciudad\_emp="Londres"**

**OR ciudad\_fab="Londres"**

**);**

1. Construir una vista con los códigos de productos suministrados ya sea por un fabricante de Londres o a una empresa de Londres.

**CREATE VIEW vista\_productos\_londres AS (**

**SELECT distinct cod\_prod**

**FROM entrega**

**WHERE cod\_fab IN (SELECT cod\_fab**

**FROM fabricante**

**WHERE ciudad\_fab="Londres")**

**OR cod\_emp IN (select cod\_emp**

**FROM empresa**

**WHERE ciudad\_emp = "Londres")**

**);**

1. Construir una vista con los códigos de empresas situadas en Londres o a las que suministre productos algún fabricante de Londres.

**CREATE VIEW vista\_empresas\_londres AS(**

**SELECT empresa.cod\_emp**

**FROM (entrega JOIN empresa ON entrega.cod\_emp=empresa.cod\_emp)**

**JOIN fabricante ON fabricante.cod\_fab=entrega.cod\_fab**

**WHERE ciudad\_emp="Londres"**

**OR ciudad\_fab="Londres"**

**);**

1. ¿Qué diferencia hay entre las tablas y las vistas creadas en los ejercicios anteriores (del 2 al 6)?

Con la base de datos EMPRESA:

1. Crear una vista **emple\_5Anios** que me obtenga el número de empleado, el nombre y la dirección de los empleados que llevan trabajando más de 5 años en la empresa.

**CREATE VIEW emple\_5Anios AS(**

**SELECT emp\_no, apellido, dir**

**FROM emple**

**WHERE fecha\_alt + INTERVAL 5 YEAR >= SYSDATE()**

**);**

1. Sobre la base de datos de mensajería, realiza una vista llamada **envios\_nombre** que nos muestre los nombres de productos, componentes, artículos y cantidades de los envíos
2. ¿Qué pasa si borramos una tabla que está siendo usada por una vista?

**Que al visualizar la vista da un problema**

1. Carga el script BDEmpresaExtra.sql.
2. Añade las restricciones de clave primaria en cada tabla. Comprueba si las que ya teníamos, emple y depart tienen claves primarias. En caso negativo, añádelas también. Pon nombre a las restricciones generadas.

**ALTER TABLE emple**

**ADD PRIMARY KEY(emp\_no)**

**ALTER TABLE depart**

**ADD PRIMARY KEY(dept\_no)**

**ALTER TABLE productos**

**ADD PRIMARY KEY(prod\_no)**

1. Añade las restricciones clave ajena (con nombre).

* La tabla clientes (ven\_no es el número de vendedor y referencia a la tabla de empleados).

**ALTER TABLE clientes**

**ADD CONSTRAINT cli\_ven\_fk FOREIGN KEY(ven\_no)**

**REFERENCES emple(emp\_no)**

**ON DELETE RESTRICT ON UPDATE RESTRICT;**

* La tabla emple (dept\_no es el número de departamento y referencia a la tabla de departamentos).

**ALTER TABLE emple**

**ADD CONSTRAINT emp\_dep\_fk FOREIGN KEY (dept\_no)**

**REFERENCES depart(dept\_no)**

**ON DELETE RESTRICT ON UPDATE RESTRICT;**

* Pedido tiene dos claves ajenas hacia productos y hacia clientes.

**ALTER TABLE pedidos**

**ALTER TABLE pedidos**

**ADD CONSTRAINT ped\_prd\_fk FOREIGN KEY(prod\_no)**

**REFERENCES productos(prod\_no)**

**ON DELETE RESTRICT ON UPDATE RESTRICT,**

**ADD CONSTRAINT ped\_cli\_fk FOREIGN KEY(cli\_no)**

**REFERENCES clientes(cli\_no)**

**ON DELETE RESTRICT ON UPDATE RESTRICT;**

Nota: Si detectas algún error en los datos al añadir las claves ajenas, haz los cambios en los datos que estimes oportunos.

1. Crear una vista en la que aparezcan los nombres de los empleados con los nombres de sus respectivos jefes.

**CREATE VIEW emple\_jefe AS(**

**SELECT e.apellido empleado, j.apellido jefe**

**FROM emple e, emple j**

**WHERE e.emp\_no = j.dir**

**);**

1. Crear una vista donde aparezcan los empleados con el nombre del departamento y la localidad en la que trabajan.

**CREATE VIEW emple\_depart AS(**

**SELECT apellido, dnombre, loc**

**FROM emple, depart**

**WHERE emple.dept\_no=depart.dept\_no**

**);**

1. Crear una vista donde se tengan los clientes con el apellido del vendedor que tienen asociado.

**CREATE VIEW cli\_emple AS(**

**SELECT c.nombre, e.apellido vendedor**

**FROM clientes c, emple e**

**WHERE c.ven\_no=e.emp\_no**

**);**

1. Crear una vista con los departamentos que estén en localidades donde existan clientes.

**CREATE VIEW depart\_existe\_cliente AS(**

**SELECT depart.dnombre**

**FROM depart**

**WHERE depart.loc IN (SELECT clientes.LOCALIDAD FROM clientes)**

**);**

1. Crear una vista con los pedidos realizados en los que aparezca el nombre del producto, la cantidad solicitada, el nombre del cliente y el apellido del vendedor que dicho cliente tiene asociado.

**CREATE VIEW ped\_cli\_vend AS(**

**SELECT descripcion, unidades, nombre, apellido**

**FROM pedidos ped, productos pro, clientes c, emple e**

**WHERE ped.prod\_no=pro.prod\_no AND ped.cli\_no=c.cli\_no AND c.ven\_no=e.emp\_no**

**);**

**CREATE VIEW v\_pedidos\_realizados AS(**

**SELECT descripcion,unidades,nombre,apellido**

**FROM ((productos JOIN pedidos ON productos.prod\_no = pedidos.prod\_no)**

**JOIN clientes ON clientes.cli\_no = pedidos.cli\_no)**

**JOIN emple ON clientes.ven\_no=emple.emp\_no  
)**

1. Crear una vista que incluya los departamentos con su nombre, la cantidad de empleados que tienen, la suma de los salarios de los empleados y la suma de las comisiones.

**CREATE VIEW CantidadEmple\_SumaSalarios\_SumaComision AS(**

**SELECT dnombre, COUNT(e.emp\_no) cantidad\_emple, SUM(IFNULL(salario,0)) suma\_salarios, SUM(IFNULL(comision,0)) suma\_comision**

**FROM depart d, emple e**

**WHERE d.dept\_no=e.dept\_no**

**GROUP BY d.dept\_no**

**);**

1. Crear una vista que incluya la cantidad de pedidos realizados por cada cliente.

**CREATE VIEW cantidad\_pedidos\_por\_cliente AS(**

**SELECT COUNT(ped\_no) cantidad\_pedidos, cli\_no**

**FROM pedidos**

**GROUP BY cli\_no);**

**CREATE VIEW v\_pedidos\_cli AS (**

**SELECT nombre, COUNT(ped\_no) cantidad\_pedidos**

**FROM pedidos p RIGHT JOIN clientes c ON p.cli\_no = c.cli\_no**

**GROUP BY c.cli\_no  
);**

1. Crear una vista con el nombre del cliente y la cantidad de dinero que ha gastado en la empresa.

**CREATE VIEW dinero\_gastado\_cli AS(**

**SELECT nombre, SUM(IFNULL(precio\*unidades,0)) dinero\_gastado**

**FROM clientes c, pedidos ped, productos prod**

**WHERE c.cli\_no=ped.cli\_no AND ped.prod\_no=prod.prod\_no**

**GROUP BY c.cli\_no**

**);**

*¡¡no salen los clientes que han gastado cero, es decir, no salen los clientes que no han realizado pedidos!!*

**CREATE VIEW dinero\_gastado\_cli AS(**

**SELECT nombre, SUM(IFNULL(precio\*unidades,0)) dinero\_gastado**

**FROM (clientes c LEFT JOIN pedidos ped ON c.cli\_no=ped.cli\_no)**

**LEFT JOIN productos prod ON ped.prod\_no=prod.prod\_no**

**GROUP BY c.cli\_no**

**);**

**CREATE VIEW dinero\_gastado\_cli AS(**

**SELECT nombre, SUM(IFNULL(precio,0)\*IFNULL(unidades,0)) dinero\_gastado**

**FROM clientes c LEFT JOIN (pedidos ped JOIN productos prod ON ped.prod\_no=prod.prod\_no) ON c.cli\_no=ped.cli\_no**

**GROUP BY c.cli\_no**

**);**

1. Crear una vista con el nombre del producto y la cantidad de unidades que se han vendido en total de cada uno de ellos.

**CREATE VIEW producto\_cantidad AS(**

**SELECT descripcion, SUM(IFNULL(unidades,0)) numero\_unidades\_perdidas  
 FROM productos LEFT JOIN pedidos on productos.prod\_no=pedidos.prod\_no**

**GROUP BY productos.prod\_no**

**);**

1. Crear una vista con el nombre del producto y la cantidad de unidades que se han vendido en total de cada uno de ellos en el último año.

**CREATE VIEW prod\_vendidos\_ultimoanyo AS(**

**SELECT descripcion, IFNULL(SUM(unidades),0) AS unidades\_perdidas**

**FROM pedidos RIGHT JOIN productos ON pedidos.prod\_no=productos.prod\_no**

**WHERE pedidos.fecha\_pedido + interval 1 year>=SYSDate()**

**GROUP BY pedidos.prod\_no**

**);**

1. Obtener los dos vendedores que han realizado más ventas.

**CREATE VIEW mejores\_vendedores AS(**

**SELECT apellido, count(ven\_no)**

**FROM emple e, clientes c, pedidos p**

**WHERE e.emp\_no=c.ven\_no AND c.cli\_no=p.cli\_no**

**GROUP BY c.ven\_no**

**ORDER BY COUNT(ven\_no) DESC**

**);**

1. Obtener los tres productos más vendidos.

**CREATE VIEW productos\_masvendidos AS(**

**SELECT descripcion, pedidos.prod\_no, SUM(unidades) unidades\_vendidas**

**FROM pedidos, productos**

**WHERE pedidos.prod\_no=productos.prod\_no**

**GROUP BY prod\_no**

**ORDER BY SUM(unidades) DESC**

**LIMIT 3**

**);**

1. Obtener los 5 productos que mayores beneficios han aportado a la empresa.
2. Obtener los dos vendedores que mayores beneficios han aportado a la empresa (los que han facturado más).

**SELECT SUM(precio\*unidades) AS precio, c.ven, apellido  
FROM ((productos pr JOIN pedidos peON p.prod\_no=pr.prod\_no) JOIN clientes c ON c.cli\_no=pe.cli\_no) JOIN emple e ON c.ven\_no=e.emp\_no  
GROUP BY e.emp\_no  
ORDER BY 1 DESC**

**LIMIT 2**

1. Obtener los 4 clientes más rentables para la empresa (los que más han gastado).

**SELECT clientes.cli\_no, clientes.nombre, sum(precio\*unidades) AS dinero\_gastado  
FROM (clientes JOIN pedidos ON pedidos.cli\_no=clientes.cli\_no) JOIN productos ON pedidos.prod\_no=productos.prod\_no  
GROUP BY clientes.cli\_no   
ORDER BY 3 DESC LIMIT 4**

1. Obtener los 2 clientes que más han comprado mesas (productos cuya descripción comienza por mesa).

**SELECT c.cli\_no, c.nombre, sum(unidades)**

**FROM (clientes c JOIN pedidos ped ON ped.cli\_no=c.cli\_no) JOIN productos pro ON ped.prod\_no=pro.prod\_no**

**WHERE pro.descripcion LIKE 'mesa%'**

**GROUP BY c.cli\_no**

**ORDER BY 3 DESC**

**LIMIT 2**

1. Obtener los 3 clientes que menos pedidos han realizado.

**SELECT c.cli\_no, c.nombre, COUNT(ped\_no) 'Pedidos realizados'  
FROM clientes c LEFT JOIN pedidos p ON p.cli\_no=c.cli\_no  
GROUP BY c.cli\_no   
ORDER BY 3 ASC   
LIMIT 3**

1. Obtener un listado de los 4 clientes que han realizado un mayor gasto en un solo pedido (sin que se repitan).

**SELECT DISTINCT c.cli\_no, c.nombre**

**FROM (clientes c JOIN pedidos pe ON pe.cli\_no=c.cli\_no)  
JOIN productos pr ON pe.prod\_no=pr.prod\_no**

**ORDER BY pr.precio\*pe.unidades DESC**

**LIMIT 4**

1. Obtener un listado de los 4 productos que han sido vendidos en mayor cantidad en un solo pedido.

**SELECT pr.descipcion, max(pe.unidades) max\_unidades**

**FROM productos pr LEFT JOIN pedidos pe ON pr.prod\_no = pes.prod\_no**

**GROUP BY pe.prod\_no**

**ORDER BY max\_unidades DESC**

**LIMIT 4**

1. Obtener los departamentos a los que pertenecen los 3 empleados más antiguos.

**SELECT e.apellido, e.dept\_no, d.dnombre, fecha\_alt**

**FROM emple e JOIN depart d ON e.dept\_no=d.dept\_no**

**ORDER BY fecha\_alt ASC**

**LIMIT 3**