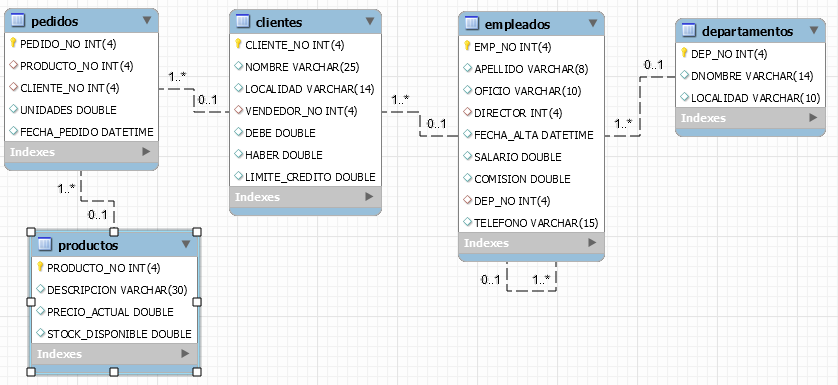


Ejecuta el script “BD-Ventas.sql” para crear la base de datos de ventas que usaremos para realizar la actividad, que estará formada por las siguientes tablas:



# Realiza, por medio del lenguaje SQL, las siguientes consultas:

1. Obtén el número de empleado, el apellido y el número de departamento de todos los empleados de la tabla empleados.

select emp\_no,apellido,dep\_no from empleados;

1. Mostrar la siguiente información de todos los empleados: apellido, la fecha de alta, el salario y el salario incrementado por una gratificación de 1000€.

select apellido,fecha\_alta,salario,salario+1000 as sal\_gratificacion from empleados;

1. Obtén todas las filas de la tabla empleados para los que en la columna OFICIO aparezca el valor 'VENDEDOR'.

select \* from empleados where oficio='VENDEDOR';

1. Visualizar la expresión formada por las columnas APELLIDO, SALARIO, COMISION, y una columna que muestre lo que gana juntando el salario y la comisión de la tabla empleados

select apellido,salario,comision,salario+ifnull(comision,0) as sal\_total from empleados;

1. APELLIDO y OFICIO de los empleados sin comisión.

select apellido,oficio from empleados where comision is null or comision=0; select apellido,oficio from empleados where ifnull(comision,0)=0;

1. Mostrar el apellido y salario de los tres empleados con mayor salario

select apellido,salario from empleados order by salario desc limit 3;

1. Empleados cuyo OFICIO sea VENDEDOR y que además su SALARIO sea estrictamente superior a 1500

select \* from empleados where oficio='VENDEDOR' and salario>1500;

1. Empleados cuyo OFICIO sea VENDEDOR o que su SALARIO sea superior o igual a 1500.

select \* from empleados where oficio='VENDEDOR' or salario>=1500;

1. Empleados cuyo **OFICIO NO sea VENDEDOR** y SALARIO sea estrictamente superior a 1500 o que sean del departamento 20.

select \* from empleados where (oficio!='VENDEDOR' and salario>1500) or dep\_no=20;

1. Muestra los Empleados de tipo vendedor y también a los de tipo director que ganen menos de 3000€.

select \* from empleados where oficio ='VENDEDOR' or (oficio='DIRECTOR' and salario<3000);

1. Obtén el total a cobrar **anualmente** por cada empleado, sabiendo que el salario almacenado es el salario mensual y que a dicho salario hay que sumarle la comisión, si la tuviera. La columna se llamará ingresos anuales.

select apellido,(salario+ifnull(comision,0))\*12 as ingresos\_anuales from empleados;

1. Mostrar una fila para cada empleado que muestre de qué trabaja cada uno de ellos con el siguiente formato: “[apellido] es [oficio]”.

select concat(apellido,' es ',oficio) from empleados;

1. Selecciona aquellos empleados cuyo apellido empiece por “M” y su salario esté comprendido entre 1.000 y

2.000 €. Visualiza su número de empleado, apellido y departamento.

select emp\_no,apellido,dep\_no from empleados

-- where apellido like 'M%' and salario between 1000 and 2000; where left(apellido,1)='M' and salario between 1000 and 2000;

1. Selecciona aquellos empleados cuyo apellido incluya una “A” en el segundo carácter.

select \* from empleados where apellido like '\_A%';

1. Seleccionar los empleados existentes en los departamentos 10 y 30.

select \* from empleados where dep\_no in (10,30);

1. Obtén todos los datos de los empleados, ordenados alfabéticamente por número del departamento y después inversamente por apellido.

select \* from empleados order by dep\_no asc, apellido desc;

1. Todos los atributos de los empleados cuya comisión no está vacía ni tampoco es 0 ordenados por comisión ascendente.

select \* from empleados where comision is not null and comision >0 order by comision asc;

1. Obtén la masa salarial mensual de todos los empleados, el salario medio y el salario máximo.

select sum(salario),avg(salario),max(salario) from empleados;

1. Obtén los salarios máximo, mínimo y la diferencia existente entre ambos.

select max(salario) as maximo,min(salario) as minimo,max(salario)-min(salario) as diferencia from empleados;

1. Obtén la fecha de alta más reciente.

select max(fecha\_alta) from empleados;

select fecha\_alta from empleados order by 1 desc limit 1;

1. Obtén los salarios medios por departamento.

select dep\_no,avg(salario) from empleados group by dep\_no;

1. Obtén cuántos empleados hay en cada departamento, ordenando el resultado en orden decreciente por el número de empleados de cada departamento.

select dep\_no,count(\*) from empleados group by dep\_no order by 2 desc;

1. Obtén el salario del empleado que mayor salario tiene dentro de cada oficio, excluyendo al presidente.

select oficio,max(salario) from empleados where oficio!='PRESIDENTE' group by oficio;

1. Seleccionar los oficios que tengan dos o más empleados.

select oficio,count(\*) from empleados group by oficio

having count(\*)>2;

1. Seleccionar los oficios que tengan dos o más empleados cuyo salario supere los 1400 €.

select oficio,count(\*) from empleados where salario>1400

group by oficio having count(\*)>2;

1. Obtén de por medio de subconsultas y con una consulta multitabla los distintos departamentos (código y nombre) en los que trabaja alguien.

select d.dep\_no,dnombre from departamentos d

where exists(select 1 from empleados e where e.dep\_no=d.dep\_no);

-- where dep\_no in (select dep\_no from empleados); select distinct d.dep\_no,dnombre

from departamentos d, empleados e where d.dep\_no=e.dep\_no;

1. Comprueba si hay algún empleado que no tenga asignado un departamento.

select \* from empleados where dep\_no is null;

1. Obtén una lista de empleados cuyo salario supere el salario medio.

select \* from empleados where salario>(select avg(salario) from empleados);

1. Visualizar el departamento con más empleados

select dep\_no,count(\*) from empleados group by dep\_no

order by 2 desc limit 1;

select dep\_no,count(\*) from empleados group by dep\_no

having count(\*)=(select max(numemps) from (select dep\_no,count(\*) as numemps from empleados group by dep\_no) Q);

1. Visualizar el departamento con más personal del oficio ‘empleado’

select dnombre,count(\*) from empleados e, departamentos d where e.dep\_no=d.dep\_no and oficio='empleado'

group by dnombre order by 2 desc limit 1;

1. Visualizar el departamento con más presupuesto asignado para pagar el salario y la comisión de sus empleados

select dep\_no from empleados group by dep\_no order by sum(salario+ifnull(comision,0)) desc limit 1;

1. Obtener información de los empleados que ganan más que cualquier empleado del departamento 30 (hacerla de 2 formas distintas).

select \* from empleados where salario+ifnull(comision,0) > all (select salario+ifnull(comision,0) from empleados where dep\_no=30);

select \* from empleados where salario+ifnull(comision,0)>(select max(salario+ifnull(comision,0)) from empleados where dep\_no=30);

select \* from empleados e1 where not exists(select 1 from empleados e2 where e2.dep\_no=30 and e1.salario+ifnull(e1.comision,0) <= e2.salario+ifnull(e2.comision,0));

1. Mostrar el código de aquellos empleados que son director. Se permite mostrar duplicados

select \* from empleados where emp\_no in (select director from empleados); select distinct director from empleados where director is not null;

1. Mostrar el código de todos los productos que se han vendido alguna vez. No se permite mostrar duplicados

select distinct producto\_no from pedidos;

1. Mostrar los siguientes datos relativos a empleados: apellido, número, nombre de departamento y localidad.

select e.apellido,e.emp\_no,d.dnombre,d.localidad

from empleados e,departamentos d where d.dep\_no=e.dep\_no;

1. Seleccionar el número, apellido y oficio de los empleados que pertenezcan al departamento de VENTAS por medio de subconsultas y por medio de una consulta multitabla.

select emp\_no,apellido,oficio from empleados

where dep\_no in (select dep\_no from departamentos where dnombre='VENTAS'); select emp\_no,apellido,oficio

from empleados e,departamentos d

where e.dep\_no=d.dep\_no and dnombre='VENTAS';

1. Visualizar los apellidos y su longitud, de los empleados cuyo departamento contenga las letras ‘ON’ en la 2ª y 3ª posición.

select e.apellido,length(apellido) from empleados e,departamentos d where e.dep\_no=d.dep\_no

and d.dnombre like '\_ON%';

1. Obtén todos los clientes junto con todos los productos que han comprado. Mostrar todos los atributos.

select c.\*,p.\*

from clientes c, productos p, pedidos ped

where c.cliente\_no=ped.cliente\_no and ped.producto\_no=p.producto\_no;

1. **Obtén todas las combinaciones posibles** de apellido de empleado con nombre de departamento. No se trata de relacionar a cada empleado con el departamento en el que efectivamente trabaja, sino de relacionar a cada empleado con cada departamento en el que podría trabajar.

select apellido,dnombre

from empleados e,departamentos d;

1. Mostar la descripción y precio de los productos vendidos por el vendedor “ALONSO”.

select p.descripcion,p.precio\_actual

from productos p, pedidos ped,clientes c,empleados e

where p.producto\_no=ped.producto\_no and ped.cliente\_no=c.cliente\_no

and c.vendedor\_no=e.emp\_no and e.apellido='ALONSO';

1. Apellido, salario y localidad de los empleados de Sevilla y Valencia

select e.apellido,e.salario,d.localidad from empleados e,departamentos d

where e.dep\_no=d.dep\_no and d.localidad in ('SEVILLA','VALENCIA');

1. Apellido, localidad y oficio de los empleados de Madrid que ganan menos de 2.000 y los de Barcelona que ganan menos de 1.500

select e.apellido,e.oficio,d.localidad from empleados e,departamentos d where e.dep\_no=d.dep\_no

and (

(d.localidad='MADRID' and e.salario<2000) or (d.localidad='BARCELONA' and e.salario<1500)

);

1. Obtén los empleados cuyo salario supera al de sus compañeros de departamento.

select \* from empleados e

where salario>(select max(salario) from empleados e2 where e.dep\_no=e2.dep\_no and e.emp\_no!=e2.emp\_no);

1. Para cada empleado indicar su apellido, número de departamento y localidad del departamento

select apellido,e.dep\_no,d.localidad from empleados e,departamentos d where e.dep\_no=d.dep\_no;

1. Para cada empleado indicar el nombre del empleado y el nombre de su jefe

select e.apellido,jefe.apellido

from empleados e join empleados jefe on (e.director=jefe.emp\_no);

1. Para cada pedido indicar el nombre del cliente y el nombre del producto comprado

select c.nombre,p.descripcion

from clientes c,pedidos ped,productos p

where c.cliente\_no=ped.cliente\_no and ped.producto\_no=p.producto\_no;

1. Obtén un listado de los productos vendidos por cada empleado (apellido del empleado y descripción y precio actual de los productos).

select distinct e.apellido,p.descripcion,p.precio\_actual from clientes c,pedidos ped,productos p,empleados e

where c.cliente\_no=ped.cliente\_no and ped.producto\_no=p.producto\_no and c.vendedor\_no=e.emp\_no;

1. Comprueba el número de veces que ha sido vendido cada producto.

select p.descripcion,count(ped.pedido\_no) as ventas,sum(ped.unidades) as unidades from pedidos ped,productos p

where ped.producto\_no=p.producto\_no group by p.descripcion;

1. Comprueba qué producto ha sido vendido un mayor número de veces.

select p.descripcion,count(ped.pedido\_no) as ventas,sum(ped.unidades) as unidades from pedidos ped,productos p

where ped.producto\_no=p.producto\_no group by p.descripcion

order by 2 desc limit 1;

1. Comprueba de qué producto se han vendido más unidades.

select p.descripcion,count(ped.pedido\_no) as ventas,sum(ped.unidades) as unidades from pedidos ped,productos p

where ped.producto\_no=p.producto\_no group by p.descripcion

order by 3 desc limit 1;

1. Calcular los ingresos medios (salario + comisión) de los empleados del departamento de VENTAS.

select avg(salario+ifnull(comision,0)) as media\_sal from empleados e,departamentos d

where e.dep\_no=d.dep\_no and d.dnombre='VENTAS';

1. Cuántos pedidos ha realizado el cliente DISTRIBUCIONES GOMEZ.

select count(p.pedido\_no) as numpedidos from clientes c,pedidos p

where c.cliente\_no=p.cliente\_no and c.nombre like '%DIST%GOM%' group by c.nombre;

1. Cuántos clientes tiene a su cargo el empleado ALONSO.

select count(c.cliente\_no) from clientes c,empleados e

where c.vendedor\_no=e.emp\_no and e.apellido='ALONSO' group by e.emp\_no;

1. Cuántos empleados tiene a su cargo (de cuántos es jefe) el empleado ALONSO.

select count(e.emp\_no)

from empleados e,empleados jefe

where e.director=jefe.emp\_no and jefe.apellido='ALONSO' group by e.director;

1. Indicar el nombre y la fecha en que se ha vendido por última vez cada producto.

select p.descripcion,max(pe.fecha\_pedido) from productos p,pedidos pe

where p.producto\_no=pe.producto\_no group by p.descripcion;

1. Para cada departamento indicar Cuánto dinero gasta en salarios, cuánto gasta en comisiones y cuánto gasta en total (salario + comisión).

select d.dnombre,sum(e.salario),sum(comision),sum(e.salario+ifnull(e.comision,0)) as total

from empleados e left outer join departamentos d on (e.dep\_no=d.dep\_no) group by d.dnombre;

1. Nombre de cada empleado que atiende a algún cliente y Cuántos clientes tiene a su cargo ese empleado. Mostrar sólo aquellos empleados que atiendan a más de 2 clientes

select e.apellido,count(c.cliente\_no) as numclientes from empleados e,clientes c

where c.vendedor\_no=e.emp\_no group by e.apellido

having count(c.cliente\_no)>2;

1. Para los productos que se han vendido alguna vez: Indicar el nombre de cada producto y Cuántas unidades se han vendido de ese producto, Mostrar sólo aquellos de los que se han vendido más de 15 unidades.

select p.descripcion,sum(pe.unidades) as unidades from productos p,pedidos pe

where p.producto\_no=pe.producto\_no group by p.descripcion

having sum(unidades)>15;

1. Indicar el nombre y la fecha en que se ha vendido por última vez cada producto siempre y cuando el importe total de ventas de ese producto sea superior a 100.

select p.descripcion,max(pe.fecha\_pedido) from productos p,pedidos pe

where p.producto\_no=pe.producto\_no group by p.descripcion

having sum(p.precio\_actual\*pe.unidades)>100;

1. Listar las localidades donde existan departamentos con empleados cuya comisión supere el 10% del salario

select distinct d.localidad

from departamentos d, empleados e

where d.dep\_no=e.dep\_no and e.comision>e.salario\*0.1;

1. Obtener el nombre de cada departamento y la media salarial de ese departamento, para los departamentos cuya media salarial es mayor que el salario del empleado 7654

select d.dnombre,avg(e.salario) as media\_salarial from departamentos d,empleados e

where d.dep\_no=e.dep\_no group by d.dnombre,d.dep\_no

having avg(e.salario)>(select salario from empleados where emp\_no=7654);

# Realiza, por medio del lenguaje SQL, las siguientes modificaciones de datos:

1. Inserta en la tabla departamentos un nuevo departamento con código 50 y sede en “SANTIAGO” que tendrá por nombre “operaciones”.

insert into departamentos (DEP\_NO,DNOMBRE,LOCALIDAD) values (50,'OPERACIONES','SANTIAGO');

1. Asigna a dicho departamento al empleado “Lopez”, que como será el jefe de ese departamento no tendrá director asignado.

update empleados set dep\_no=50, director=null where apellido='LOPEZ'; select \* from empleados;

1. Inserta con los códigos 8000, 8001 y 8002 a los empleados “MARTINEZ”, “PEREZ” y “RODRIGUEZ”, analistas que acaban de ser contratados por el departamento de operaciones. Sus salarios serán respectivamente de 1000, 1100 y 1200€, sin comisiones. Su director será el 7521.

insert into empleados (EMP\_NO,APELLIDO,OFICIO,DIRECTOR,FECHA\_ALTA,SALARIO,COMISION,DEP\_NO, TELEFONO ) values(8000,'MARTINEZ','ANALISTA',7521,date(now()),1000,null,50,null),

(8001,'PEREZ','ANALISTA',7521,date(now()),1100,null,50,null), (8002,'RODRIGUEZ','ANALISTA',7521,date(now()),1200,null,50,null);

1. Asignar una comisión de 2.000€ a los vendedores que no ganan nada en concepto de comisión

update empleados set comision=2000 where comision is null and oficio='VENDEDOR';

1. Actualizar con una sola operación el pedido 1000 de modo que aumente en 1 unidad el producto comprado y se reduzca en una unidad el stock disponible del producto vendido.

update pedidos ped,productos prod set ped.unidades=ped.unidades+1,prod.STOCK\_DISPONIBLE=prod.STOCK\_DISPONIBLE-1 where ped.producto\_no=prod.producto\_no and ped.pedido\_no=1000;

1. El pedido 1001 no ha sido pagado, por ese motivo debemos añadir al debe del cliente el importe del pedido.

update clientes cl,pedidos ped,productos prod set debe=debe+(prod.precio\_actual\*ped.unidades)

where cl.CLIENTE\_NO=ped.CLIENTE\_NO and ped.PRODUCTO\_NO=prod.producto\_no and ped.PEDIDO\_NO=1001;

1. Inserta un nuevo departamento con código 0, nombre “DESCONOCIDO” y sede “DESCONOCIDA”.

insert into departamentos (DEP\_NO,DNOMBRE,LOCALIDAD) values (0,'DESCONOCIDO','DESCONOCIDA');

1. Copia todos los empleados entre el 7839 y el 7876, poniéndoles como código el código actual menos 7838. Su fecha de contratación debe ser la fecha actual.

insert into empleados (EMP\_NO,APELLIDO,OFICIO,DIRECTOR,FECHA\_ALTA,SALARIO,COMISION,DEP\_NO, TELEFONO ) select EMP\_NO-7838,APELLIDO,OFICIO,DIRECTOR,date(now()),SALARIO,COMISION,DEP\_NO, TELEFONO

from empleados where emp\_no between 7839 and 7876;

1. Crea una nueva tabla de “EMP\_CONT” con la misma estructura que la tabla de empleados, e inserta en ella los empleados del departamento de contabilidad.

create table emp\_cont as select \* from empleados where dep\_no=(select dep\_no from departamentos where dnombre='CONTABILIDAD');

1. Sube el salario un 2% a los empleados con oficio vendedor y súbelo un 3% a los que no son vendedor ni director (con una sola instrucción).

update empleados set salario=salario\*(case when oficio='VENDEDOR' then 1.02 when oficio not in ('VENDEDOR','DIRECTOR') then 1.03 else 1 end);

1. Crea una nueva tabla PUESTOS(codpuesto,puesto). Inserta en dicha tabla los distintos puestos que ocupan los empleados (sin repetir).

create table puestos(codpuesto smallint auto\_increment primary key, puesto varchar(20));

insert into puestos (puesto) select distinct oficio from empleados;

1. Crea un campo codpuesto en la tabla de empleado y asigna a cada empleado el código de puesto que le corresponda en función de su oficio. Elimina posteriormente el campo OFICIO de la tabla de empleados.

alter table empleados add codpuesto smallint; update empleados e, puestos p

set e.codpuesto=p.codpuesto where e.oficio=p.puesto; alter table empleados drop oficio;

1. A todos los empleados que no tengan asignado un departamento, asígnales el departamento desconocido.

update empleados set dep\_no=(select dep\_no from departamentos where dnombre='DESCONOCIDO') where dep\_no is null;

1. El empleado 7499 cambiará de lugar de trabajo, pasando al departamento CONTABILIDAD con el puesto de EMPLEADO. Modifica ambos campos con una sola sentencia.

update empleados

set dep\_no=(select dep\_no from departamentos where dnombre='CONTABILIDAD'), codpuesto=(select codpuesto from puestos where puesto='EMPLEADO')

where emp\_no=7499;

1. Transfiere a todos los empleados del departamento 30 que no tengan comisión al departamento 40, y asígnales una comisión de 200€.

update empleados set dep\_no=40, comision=200

where dep\_no=30 and comision is null;

1. Asigna una comisión de 500 a todos los empleados que no teniendo comisión tengan un salario superior a 2500€.

update empleados set comision=500

where comision is null and salario>2500;

1. Asigna una comisión del 10% de su salario a todos los empleados que no teniendo comisión tengan un salario inferior a 2500€.

update empleados set comision=salario\*0.1 where comision is null and salario<2500;

1. Baja todos los salarios un 10% e incrementa la comisión un 20%.

update empleados set salario=salario\*0.9,COMISION=comision\*1.2;

1. Elimina todos los empleados de la tabla EMP\_CONT cuyo código sea mayor que 10.

delete from emp\_cont where emp\_no>10;

1. Despide (elimina) a aquellos empleados cuya comisión supera a su salario.

delete from empleados where comision>salario;

1. Jubila (elimina) a aquellos empleados contratados antes del 1 de marzo de 1981.

delete from empleados where FECHA\_ALTA<'19810301';

1. Borra el departamento CONTABILIDAD. Antes de eliminarlo, tendrás que reubicar a sus empleados y asignarlos al departamento DECONOCIDO.

update empleados set DEP\_NO=(select dep\_no from departamentos where dnombre='DESCONOCIDO')

where dep\_no in (select dep\_no from departamentos where dnombre='CONTABILIDAD'); delete from departamentos where dnombre='CONTABILIDAD';

1. Elimina todos los empleados del departamento de ventas cuyo apellido contenga como segunda letra una A, que tengan menos de 7 caracteres y que no terminen en O.

delete e from empleados e, departamentos d

where e.DEP\_NO=d.dep\_no and d.DNOMBRE='VENTAS' and e.APELLIDO like '\_A%' and length(e.apellido)<7 and e.apellido not like '%O';

1. Añade un campo SEDE de tipo smallint a la tabla de departamentos.

alter table departamentos add sede smallint;

1. Asigna la sede 1 a todos los departamentos excepto al desconocido, que tendrá sede 0.

update departamentos set sede=if(DNOMBRE='CONTABILIDAD',0,1);

1. Duplica todos los departamentos de la sede 1 para crearlos también en la sede 2. Los códigos de los nuevos departamentos se obtendrán sumándole 50 al código de departamento que estén copiando.

insert into departamentos (DEP\_NO,DNOMBRE,LOCALIDAD,sede)

select DEP\_NO+50,DNOMBRE,LOCALIDAD,2 from departamentos where sede=1;