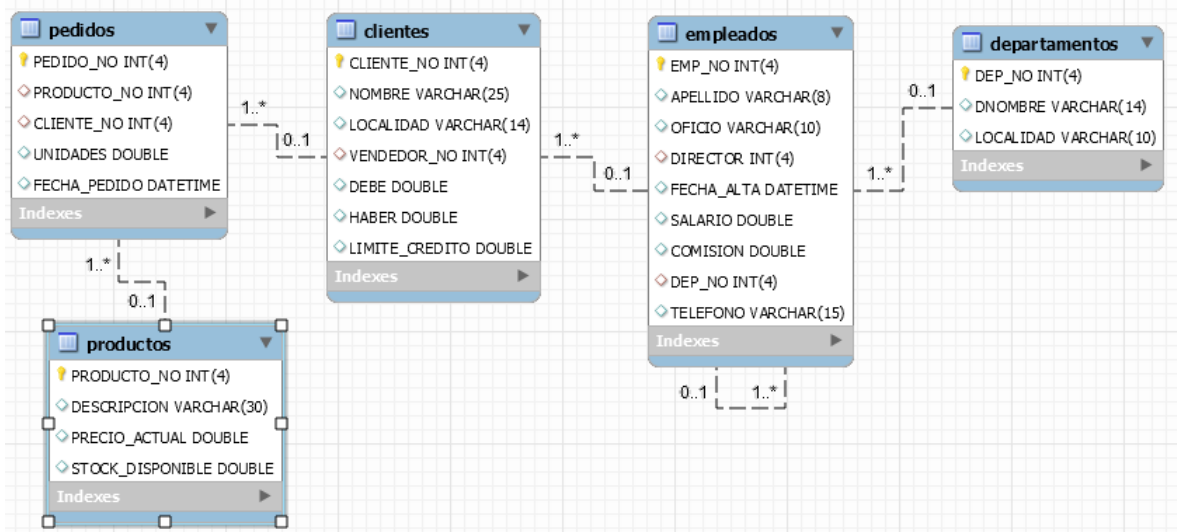


## Ejercicios DML

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;
```

2. 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;
```

3. 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';
```

4. 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;
```

5. 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;
```

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

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

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

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

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

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

9. 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;
```

10. 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);
```

11. 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;
```

12. 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;
```

13. 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;
```

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

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

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

```
select * from empleados where dep_no in (10,30);
```

16. 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;
```

17. 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;
```

18. 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;
```

19. 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;
```

20. Obtén la fecha de alta más reciente.

```
select max(fecha_alta) from empleados;
select fecha_alta from empleados order by 1 desc limit 1;
```

21. Obtén los salarios medios por departamento.

```
select dep_no,avg(salario) from empleados group by dep_no;
```

22. 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;
```

23. 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;
```

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

```
select oficio,count(*) from empleados
group by oficio
having count(*)>2;
```

25. 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;
```

26. 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;
```

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

```
select * from empleados where dep_no is null;
```

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

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

29. 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);
```

30. 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;
```

31. 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;
```

32. 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));
```

33. 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;
```

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

```
select distinct producto_no from pedidos;
```

35. 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;
```

36. 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';
```

37. 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%';
```

38. 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;
```

39. **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;
```

40. Mostrar 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';
```

41. 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');
```

42. 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)
);
```

43. 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);
```

44. 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;
```

45. 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);
```

46. 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;
```

47. 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;
```

48. 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;
```

49. 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;
```

50. 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;
```

51. 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';
```

52. 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;
```

53. 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;
```

54. 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;
```

55. 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;
```

56. 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;
```

57. 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;
```

58. 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;
```

59. 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;
```

60. 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;
```

61. 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:**

62. 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');
```

63. 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;
```

64. 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);
```

65. 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';
```

66. 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;
```

67. 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;
```

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

```
insert into departamentos (DEP_NO,DNOMBRE,LOCALIDAD) values
(0,'DESCONOCIDO','DESCONOCIDA');
```

69. 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;
```

70. 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');
```

71. 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);
```

72. 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;
```

73. 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;
```

74. 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;
```

75. 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;
```

76. 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;
```

77. 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;
```

78. 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;
```

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

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

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

```
delete from emp_cont where emp_no>10;
```

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

```
delete from empleados where comision>salario;
```

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

```
delete from empleados where FECHA_ALTA<'19810301';
```

83. 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';
```

84. 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';
```

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

```
alter table departamentos add sede smallint;
```

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

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

87. 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;
```