

```

1 • CREATE DATABASE IF NOT EXISTS `empleadoss_departamentoss` /*!40100 DEFAULT CHARACTER SET utf8 COLLATE utf8_bin */;
2 • USE `empleadoss_departamentoss`;
3
4 • /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
5 • /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
6 • /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
7 • /*!40101 SET NAMES utf8 */;
8 • /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
9 • /*!40103 SET TIME_ZONE='+00:00' */;
10 • /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
11 • /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
12 • /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
13 • /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
14 --
15 -- Table structure for table `departamentos`
16 --
17
18 • DROP TABLE IF EXISTS `departamentoss`;
19 • /*!40101 SET @saved_cs_client      = @@character_set_client */;
20 • /*!40101 SET character_set_client = utf8 */;
21 • CREATE TABLE `departamentos` (
22     `codDepto` varchar(4) COLLATE utf8_bin NOT NULL,
23     `nombreDpto` varchar(20) COLLATE utf8_bin NOT NULL,
24     `ciudad` varchar(15) COLLATE utf8_bin DEFAULT NULL,
25     `codDirector` varchar(12) COLLATE utf8_bin DEFAULT NULL,
26     PRIMARY KEY (`codDepto`),
27     KEY `FK_EmpDir` (`codDirector`),
28     CONSTRAINT `FK_EmpDir` FOREIGN KEY (`codDirector`) REFERENCES `empleadoss` (`jefeID`)
29 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
30 • /*!40101 SET character_set_client = @saved_cs_client */;

```

```
33 -- Dumping data for table `departamentos`
34 --
35
36 • LOCK TABLES `departamentos` WRITE;
37 • /*!40000 ALTER TABLE `departamentos` DISABLE KEYS */;
38 • INSERT INTO `departamentos` VALUES ('1000','GERENCIA','CIUDAD REAL','31.840.269');
39 • INSERT INTO `departamentos` VALUES ('1500','PRODUCCIÓN','CIUDAD REAL','16.211.383');
40 • INSERT INTO `departamentos` VALUES ('2000','VENTAS','CIUDAD REAL','31.178.144');
41 • INSERT INTO `departamentos` VALUES ('2100','VENTAS','BARCELONA','16.211.383');
42 • INSERT INTO `departamentos` VALUES ('2200','VENTAS','VALENCIA','16.211.383');
43 • INSERT INTO `departamentos` VALUES ('2300','VENTAS','MADRID','16.759.060');
44 • INSERT INTO `departamentos` VALUES ('3000','INVESTIGACIÓN','CIUDAD REAL','16.759.060');
45 • INSERT INTO `departamentos` VALUES ('3500','MERCADEO','CIUDAD REAL','22.222.222');
46 • INSERT INTO `departamentos` VALUES ('4000','MANTENIMIENTO','CIUDAD REAL','333.333.333');
47 • INSERT INTO `departamentos` VALUES ('4100','MANTENIMIENTO','BARCELONA','16.759.060');
48 • INSERT INTO `departamentos` VALUES ('4200','MANTENIMIENTO','VALENCIA','16.759.060');
49 • INSERT INTO `departamentos` VALUES ('4300','MANTENIMIENTO','MADRID','16.759.060');
50 • /*!40000 ALTER TABLE `departamentos` ENABLE KEYS */;
51 • UNLOCK TABLES;
```

```
57 • DROP TABLE IF EXISTS `empleados`;
58 • /*!40101 SET @saved_cs_client      = @@character_set_client */;
59 • /*!40101 SET character_set_client = utf8 */;
60 • ☹ CREATE TABLE `empleados` (
61     `nDIEmp` varchar(12) COLLATE utf8_bin NOT NULL,
62     `nomEmp` varchar(30) COLLATE utf8_bin NOT NULL,
63     `sexEmp` char(1) COLLATE utf8_bin NOT NULL,
64     `fecNac` date NOT NULL,
65     `fecIncorporacion` date NOT NULL,
66     `salEmp` float NOT NULL,
67     `comisionE` float NOT NULL,
68     `cargoE` varchar(15) COLLATE utf8_bin NOT NULL,
69     `jefeID` varchar(12) COLLATE utf8_bin DEFAULT NULL,
70     `codDepto` varchar(4) COLLATE utf8_bin NOT NULL,
71     PRIMARY KEY (`nDIEmp`),
72     KEY `FK_Empl` (`jefeID`),
73     KEY `FK_Dpto` (`codDepto`),
74     CONSTRAINT `FK_Dpto` FOREIGN KEY (`codDepto`) REFERENCES `departamentos` (`codDepto`),
75     CONSTRAINT `FK_Empl` FOREIGN KEY (`jefeID`) REFERENCES `empleados` (`nDIEmp`)
76 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
77 • /*!40101 SET character_set_client = @saved_cs_client */;
```

```

83 • LOCK TABLES `empleados` WRITE;
84 • /*!40000 ALTER TABLE `empleados` DISABLE KEYS */;
85 • INSERT INTO `empleados` VALUES ('1.130.222','José Giraldo','M','1985-01-20','2000-11-01',1200000,400000,'Asesor','22.222.222','3500');
86 • INSERT INTO `empleados` VALUES ('1.130.333','Pedro Blanco','M','1987-10-28','2000-10-01',800000,3000000,'Vendedor','31.178.144','2000');
87 • INSERT INTO `empleados` VALUES ('1.130.444','Jesús Alfonso','M','1988-03-14','2000-10-01',800000,3500000,'Vendedor','31.178.144','2000');
88 • INSERT INTO `empleados` VALUES ('1.130.555','Julián Mora','M','1989-07-03','2000-10-01',800000,3100000,'Vendedor','31.178.144','2200');
89 • INSERT INTO `empleados` VALUES ('1.130.666','Manuel Millán','M','1990-12-08','2004-06-01',800000,3700000,'Vendedor','31.178.144','2300');
90 • INSERT INTO `empleados` VALUES ('1.130.777','Marcos Cortez','M','1986-06-23','2000-04-16',2550000,500000,'Mecánico','333.333.333','4000');
91 • INSERT INTO `empleados` VALUES ('1.130.782','Antonio Gil','M','1980-01-23','2010-04-16',850000,1500000,'Técnico','16.211.383','1500');
92 • INSERT INTO `empleados` VALUES ('1.751.219','Melissa Roa','F','1960-06-19','2001-03-16',2250000,2500000,'Vendedor','31.178.144','2100');
93 • INSERT INTO `empleados` VALUES ('11.111.111','Irene Díaz','F','1979-09-28','2004-06-01',1050000,200000,'Mecánico','333.333.333','4200');
94 • INSERT INTO `empleados` VALUES ('16.211.383','Luis Pérez','M','1956-02-25','2000-01-01',5050000,0,'Director','31.840.269','1500');
95 • INSERT INTO `empleados` VALUES ('16.759.060','Darío Casas','M','1960-04-05','1992-11-01',4500000,500000,'Investigador','31.840.269','3000');
96 • INSERT INTO `empleados` VALUES ('19.709.802','William Daza','M','1982-10-09','1999-12-16',2250000,1000000,'Investigador','16.759.060','3000');
97 • INSERT INTO `empleados` VALUES ('22.222.222','Carla López','F','1975-05-11','2005-07-16',4500000,500000,'Jefe Mercadeo','31.840.269','3500');
98 • INSERT INTO `empleados` VALUES ('22.222.333','Carlos Roza','M','1975-05-11','2001-09-16',750000,500000,'Vigilante','31.840.269','3500');
99 • INSERT INTO `empleados` VALUES ('31.174.099','Diana Solarte','F','1957-11-19','1990-05-16',1250000,500000,'Secretaria','31.840.269','1000');
100 • INSERT INTO `empleados` VALUES ('31.178.144','Rosa Angulo','F','1957-03-15','1998-08-16',3250000,3500000,'Jefe Ventas','31.840.269','2000');
101 • INSERT INTO `empleados` VALUES ('31.840.269','María Rojas','F','1959-01-15','1990-05-16',6250000,1500000,'Gerente',NULL,'1000');
102 • INSERT INTO `empleados` VALUES ('333.333.333','Elisa Rojas','F','1979-09-28','2004-06-01',3000000,1000000,'Jefe Mecánicos','31.840.269','4000');
103 • INSERT INTO `empleados` VALUES ('333.333.334','Marisol Pulido','F','1979-10-01','1990-05-16',3250000,1000000,'Investigador','16.759.060','3000');
104 • INSERT INTO `empleados` VALUES ('333.333.335','Ana Moreno','F','1992-01-05','2004-06-01',1200000,400000,'Secretaria','16.759.060','3000');
105 • INSERT INTO `empleados` VALUES ('333.333.336','Carolina Ríos','F','1992-02-15','2000-10-01',1250000,500000,'Secretaria','16.211.383','1500');
106 • INSERT INTO `empleados` VALUES ('333.333.337','Edith Muñoz','F','1992-03-31','2000-10-01',800000,3600000,'Vendedor','31.178.144','2100');
107 • INSERT INTO `empleados` VALUES ('444.444','Abel Gómez','M','1939-12-24','2000-10-01',1050000,200000,'Mecánico','333.333.333','4300');
108 • INSERT INTO `empleados` VALUES ('737.689','Mario Llano','M','1945-08-30','1990-05-16',2250000,2500000,'Vendedor','31.178.144','2300');
109 • INSERT INTO `empleados` VALUES ('768.782','Joaquín Rosas','M','1947-07-07','1990-05-16',2250000,2500000,'Vendedor','31.178.144','2200');
110 • INSERT INTO `empleados` VALUES ('888.888','Iván Duarte','M','1955-08-12','1998-05-16',1050000,200000,'Mecánico','333.333.333','4100');
111 • /*!40000 ALTER TABLE `empleados` ENABLE KEYS */;
112 • UNLOCK TABLES;

```

```
123      -- Dump completed
124
125      # 1. Obtener los datos completos de los empleados.
126
127      • SELECT * FROM empleados;
128
129      # 2. Obtener los datos completos de los departamentos.
130
131      • SELECT * FROM departamentos;
132
133      # 3. Obtener los datos de los empleados con cargo 'Secretaria'.
134
135      • SELECT * FROM empleados
136        WHERE cargoE = 'Secretaria';
137
138      # 4. Obtener el nombre y salario de los empleados.
139
140      • SELECT nomEmp, salEmp FROM empleados;
141
142      # 5. Obtener los datos de los empleados vendedores, ordenado por nombre.
143
144      • SELECT * FROM empleados
145        WHERE cargoE = 'Vendedor'
146        ORDER BY nomEmp ASC;
147
148      # 6. Listar el nombre de los departamentos
149
150      • SELECT DISTINCT nombreDpto FROM departamentos;
151
```

# 7. Obtener el nombre y cargo de todos los empleados, ordenado por salario

```
SELECT nomEmp, cargoE FROM empleados  
ORDER BY salEmp;
```

# 8. Listar los salarios y comisiones de los empleados del departamento 2000, ordenado por comisión

```
SELECT salEmp, comisionE FROM empleados  
WHERE codDepto = '2000'  
ORDER BY comisionE ASC;
```

# 9. Listar todas las comisiones

```
SELECT comisionE FROM empleados;
```

# 10. Obtener el valor total a pagar que resulta de sumar a los empleados del departamento 3000 una bonificación de 500.000, en orden alfabético del empleado

```
SELECT nomemp, salemp, (salemp+500000) AS 'pago estimado' FROM empleados  
WHERE codDepto = '3000'  
ORDER BY nomemp ASC;
```

# 11. Obtener la lista de los empleados que ganan una comisión superior a su sueldo.

```
SELECT nomEmp, salEmp, comisionE FROM empleados  
WHERE salEmp < comisionE;
```

# 12. Listar los empleados cuya comisión es menor o igual que el 30% de su sueldo.

```
SELECT nomEmp, salEmp, comisionE, (salEmp * 0.3) AS porcionsueldo FROM empleados  
WHERE comisionE <= (salEmp * 0.3)
```

# 13. Elabore un listado donde para cada fila, figure 'Nombre' y 'Cargo' antes del valor respectivo para cada empleado.

```
SELECT nomemp AS 'Nombre', cargoe AS 'Cargo'
FROM empleados;
```

# 14. Hallar el salario y la comisión de aquellos empleados cuyo número de documento de identidad es superior al '19.709.802'

```
SELECT ndiemp, salemp, comisionE FROM empleados
WHERE ndiemp > '19.709.802';
```

# 15. Muestra los empleados cuyo nombre empiece entre las letras J y Z (rango). Liste estos empleados y su cargo por orden alfabético.

```
SELECT nomemp, cargoe FROM empleados
WHERE lower(nomemp) >= 'j' AND lower(nomemp) < 'z'
ORDER BY nomemp;
```

# 16. Listar el salario, la comisión, el salario total (salario + comisión), documento de identidad del  
# empleado y nombre, de aquellos empleados que tienen comisión superior a 1.000.000, ordenar el  
# informe por el número del documento de identidad.

```
SELECT nomemp ,salemp, comisione, (salemp + comisione) AS sueldoTotal, ndiemp FROM empleados
WHERE comisione > '1.000.000'
ORDER BY ndiemp;
```

# 17. Obtener un listado similar al anterior, pero de aquellos empleados que NO tienen comisión.

```
SELECT nomemp ,salemp, comisione, (salemp + comisione) AS sueldoTotal, ndiemp FROM empleados
WHERE comisione = 0
ORDER BY ndiemp;
```

# 18. Hallar los empleados cuyo nombre no contiene la cadena "MA".

```
SELECT * FROM empleados
WHERE lower(nomEmp) NOT LIKE '%na%';
```

# 19. Obtener los nombres de los departamentos que no sean "Ventas" ni "Investigación" NI 'MANTENIMIENTO', ordenados por ciudad...

```
SELECT nombredpto, ciudad FROM departamentos
WHERE NOT nombredpto = 'INVESTIGACIÓN' AND NOT nombredpto = 'VENTAS' AND NOT nombredpto = 'MANTENIMIENTO'
ORDER BY ciudad ASC;
```

```
SELECT nombreDpto, ciudad
FROM departamentos
WHERE nombreDpto NOT IN ('VENTAS', 'INVESTIGACIÓN', 'MANTENIMIENTO')
ORDER BY ciudad ASC;
```

#20. Obtener el nombre y el departamento de los empleados con cargo 'Secretaria' o 'Vendedor', que no trabajan en el departamento de "PRODUCCION",  
# cuyo salario es superior a \$1.000.000, ordenados por fecha de incorporación.

```
SELECT cargoe FROM empleados;

SELECT e.nomemp, d.nombreDpto
FROM empleados e , departamentos d
WHERE e.codDepto=d.codDepto AND e.cargoE = 'Secretaria' AND e.cargoE = 'VENDEDOR';
```

# 21. Obtener información de los empleados cuyo nombre tiene exactamente 11 caracteres.

```
SELECT * FROM empleados
WHERE char_length(nomEmp) = 11;
```



# 22. Obtener información de los empleados cuyo nombre tiene al menos 11 caracteres.

```
SELECT * FROM empleados
WHERE char_length(nomEmp) >= 11;
```

# 23. Listar los datos de los empleados cuyo nombre inicia por la letra 'M', su salario es mayor a \$800.000 o reciben comisión y trabajan para el departamento de 'VENTAS'.

```
SELECT * FROM empleados e , departamentos d
WHERE e.codDepto = d.codDepto AND e.nomEmp LIKE 'M%' AND (e.salEmp > 800.000 OR e.comisionE > 0) AND lower(d.nombreDpto) = 'ventas' ;
```

# 24. Obtener los nombres, salarios y comisiones de los empleados que reciben un salario situado entre la mitad de la comisión la propia comisión.

```
SELECT nomemp, comisione, salemp, (comisione/2) FROM empleados
WHERE salemp BETWEEN (comisione/2) AND comisione;
```

# 25. Mostrar el salario más alto de la empresa.

```
SELECT MAX(salemp) FROM empleados;
```

# 26. Mostrar cada una de las comisiones y el número de empleados que las reciben. Solo si tiene comision.

```
SELECT comisione, count(comisione) FROM empleados
WHERE comisione > 0
GROUP BY comisione;
```

```
SELECT comisione, COUNT(*) FROM empleados
WHERE comisione > 0
GROUP BY comisione;
```

# 27. Mostrar el nombre del último empleado de la lista por orden alfabético.

```
SELECT MAX(nomemp) FROM empleados;
```

# 28. Hallar el salario más alto, el más bajo y la diferencia entre ellos.

```
SELECT MAX(salemp), MIN(salemp), MAX(salemp)-MIN(salemp) AS DIFERENCIA FROM empleados;
```

# 29. Mostrar el número de empleados de sexo femenino y de sexo masculino, por departamento.

```
SELECT coddepto, sexemp, count(sexemp) as Cantidad FROM empleados  
GROUP BY coddepto, sexemp;
```

# 30. Hallar el salario promedio por departamento.

```
SELECT codDepto, AVG(salemp) AS media  
FROM empleados  
GROUP BY codDepto;
```

# 31. Mostrar la lista de los empleados cuyo salario es mayor o igual que el promedio de la empresa. Ordenarlo por departamento.

```
SELECT CodDepto, nomemp FROM empleados  
WHERE salemp >= (SELECT AVG(salemp)  
                FROM empleados)  
ORDER BY CodDepto;
```

# 32. Hallar los departamentos que tienen más de tres empleados. Mostrar el número de empleados de esos departamentos.

```
select d.codDepto, d.nombreDpto, count(*) as 'Num empleados'
from departamentos d, empleados e
where d.codDepto=e.codDepto
group by d.codDepto
having count(*) >= 3;
```

# 33. Mostrar el código y nombre de cada jefe, junto al número de empleados que dirige. Solo los que tengan mas de dos empleados (2 incluido).

```
select j.nDIEmp, j.nomEmp, count(*) as 'Num Empleados'
from empleados e, empleados j
where e.jefeID=j.nDIEmp
group by j.nDIEmp2
having count(*)>=2
order by count(*) desc;
```

# 34. Hallar los departamentos que no tienen empleados.

```
SELECT d.codDepto, d.nombreDpto
FROM empleados e, departamentos d
WHERE e.codDepto = d.codDepto
GROUP BY d.codDepto
HAVING COUNT(nomEmp) = 0;
```

# 35. Mostrar el nombre del departamento cuya suma de salarios sea la más alta, indicando el valor de la suma.

```
SELECT d.nombreDpto, SUM(e.salEmp)
FROM empleados e, departamentos d
WHERE e.codDepto = d.codDepto
GROUP BY d.nombreDpto
```