|  |  |  |
| --- | --- | --- |
|  | | |
|  | **Hurlingham**, Agosto del 2021  Buenos Aires, Argentina | |
| **ESTRATEGIAS DE PERSISTENCIA (759\_45)**  C2-2021  Profesor: Lic. Pablo Marcelli  **TP\_CLASE1\_PRACTICA**  Alumno: Raul A. Gutierrez  correo electronico: [correo@raulgutierrez.com.ar](mailto:correo@raulgutierrez.com.ar)  [raul.gutierrez@estudiantes.unahur.edu.ar](mailto:raul.gutierrez@estudiantes.unahur.edu.ar)  Entrega: Lunes 16 de Agosto 2021 | |  |
|  |  |  |

Ejecicio 1

**GENRACION DEL MODELO DE TABLAS**

CREATE DATABASE IF NOT EXISTS `DBRRHH-sueldos`;

USE `DBRRHH-sueldos`;

CREATE TABLE IF NOT EXISTS `departamentos` (

`codDepto` VARCHAR(4),

`nombreDpto` VARCHAR(20),

`ciudad` VARCHAR(15),

`codDirector` VARCHAR(12),

PRIMARY KEY (`codDepto`)

);

CREATE TABLE IF NOT EXISTS `empleados` (

`nDiEmp` VARCHAR(12) NOT NULL,

`nomEmp` VARCHAR(30),

`sexEmp` VARCHAR(1),

`fecNac` DATE,

`fecIncorporacion` DATE,

`salEmp` FLOAT,

`comisionE` FLOAT,

`cargoE` VARCHAR(15),

`jefeID` VARCHAR(12),

`codDepto` VARCHAR(4),

PRIMARY KEY (`nDiEmp`),

CONSTRAINT `FK\_jefeID` FOREIGN KEY (`jefeID`) REFERENCES `empleados` (`nDIEmp`),

CONSTRAINT `FK\_codDepto` FOREIGN KEY (`codDepto`) REFERENCES `departamentos` (`codDepto`)

);

**Ajuste de registros**

Se modificaron registros necesarios para la vinculacion por claves primarias y utilizacion de feraneas.

ALTER TABLE `departamentos`

ADD CONSTRAINT `FK\_codDirector` FOREIGN KEY (`codDirector`) REFERENCES `empleados` (`nDiEmp`)

;

**Insercion de registros masiva**

INSERT INTO `departamentos` (`codDepto`, `nombreDpto`, `ciudad`, `codDirector`) VALUES ('0001', 'Informatica', 'BsAs', NULL);

INSERT INTO `departamentos` (`codDepto`, `nombreDpto`, `ciudad`, `codDirector`) VALUES ('0002', 'Administracion', 'CABA', NULL);

INSERT INTO `departamentos` (`codDepto`, `nombreDpto`, `ciudad`, `codDirector`) VALUES ('0003', 'Expedicion', 'Rosario', NULL);

INSERT INTO `empleados` (`nDiEmp`, `nomEmp`, `sexEmp`, `fecNac`, `fecIncorporacion`, `salEmp`, `comisionE`, `cargoE`, `jefeID`, `codDepto`) VALUES ('1001', 'JUAN MARCELO MENDEZ', 'M', '1971-04-06', '2020-01-03', '60000', '1000', 'Operaciones', NULL, '0003'), ('1002', 'GRACIELA INES MORLA', 'F', '1979-04-28', '2018-08-01', '75000', '1000', 'Sistemas', NULL, '0001'), ('1003', 'CARLOS ARDINAN', 'M', '1966-09-29', '2016-03-01', '66000', '1000', 'Contador', NULL, '0002');

INSERT INTO `empleados` (`nDiEmp`, `nomEmp`, `sexEmp`, `fecNac`, `fecIncorporacion`, `salEmp`, `comisionE`, `cargoE`, `jefeID`, `codDepto`) VALUES ('1004', 'RODOLFO MONTENEGRO', 'M', '1987-10-23', '2020-04-03', '66000', '1000', 'Sistemas', NULL, '0001'), ('1005', 'CARLA BIZICONTI', 'F', '1999-06-09', '2020-06-10', '66000', '1000', 'Administracion', NULL, '0002');

1.- Obtener los datos completos de los empleados.

SELECT empleados.nDIEmp AS 'Codigo empleado',

empleados.nomEmp AS 'Nombre empleado',

empleados.sexEmp AS 'Sexo',

empleados.fecNac AS 'Fecha nacimiento',

empleados.fecIncorporacion AS 'Fecha incorpora.',

empleados.salEmp AS 'Salario',

empleados.comisionE AS 'Comision',

empleados.cargoE AS 'Cargo',

empleados.jefeID AS 'codigo del Jefe',

empleados.codDepto AS 'codigo Depto.'

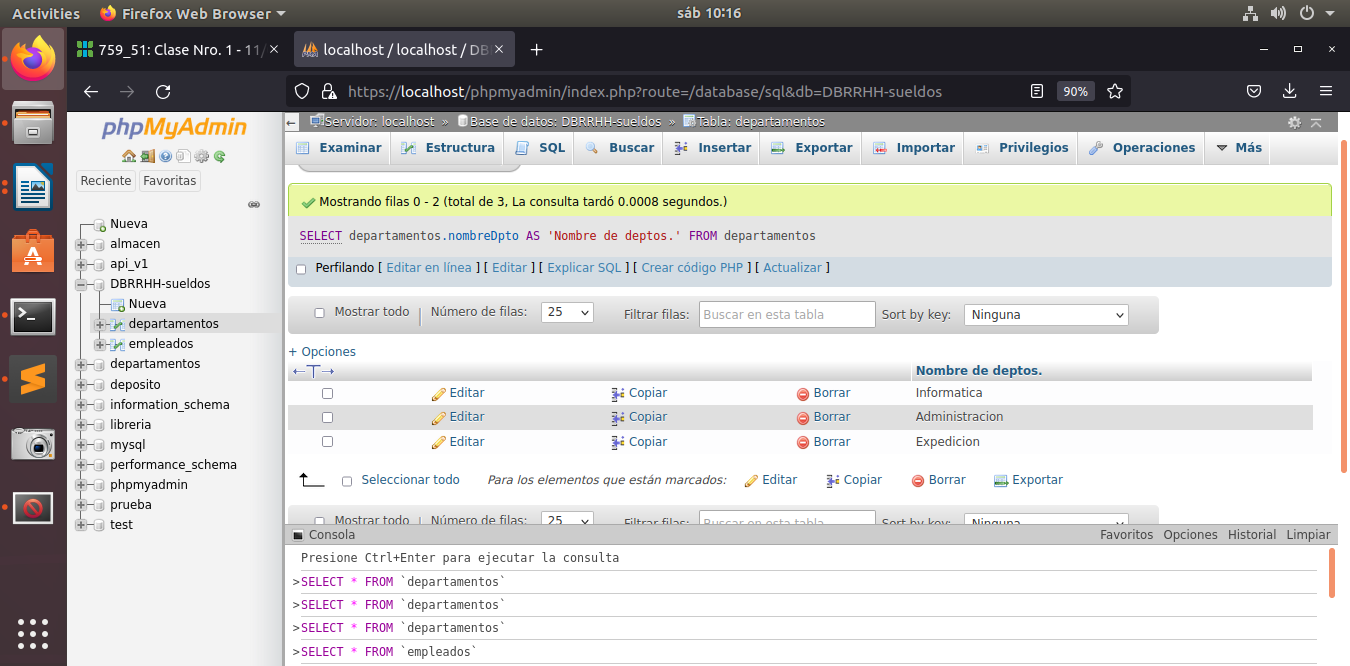
FROM empleados



2.- Listar el nombre de los departamentos.

SELECT departamentos.nombreDpto AS 'Nombre de deptos.'

FROM departamentos



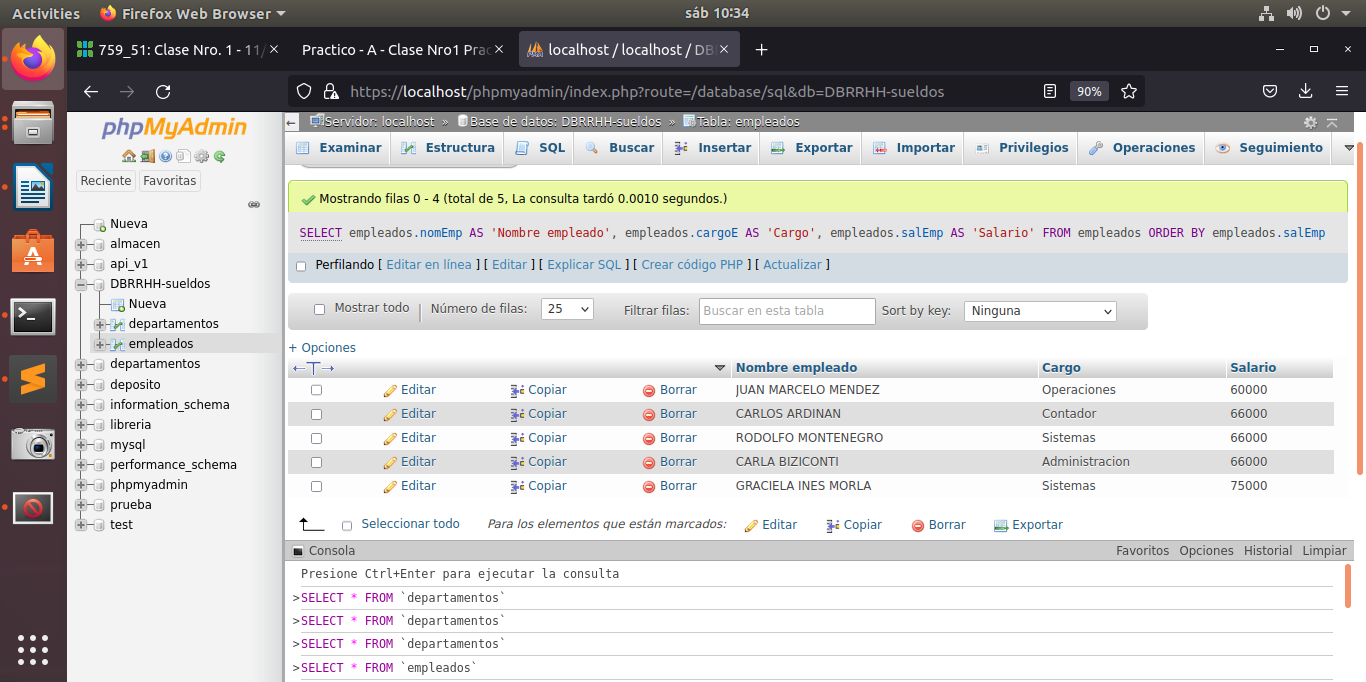
3.- Obtener el nombre y cargo de todos los empleados, ordenado por salario.

SELECT empleados.nomEmp AS 'Nombre empleado',

empleados.cargoE AS 'Cargo', empleados.salEmp AS 'Salario'

FROM empleados

ORDER BY empleados.salEmp



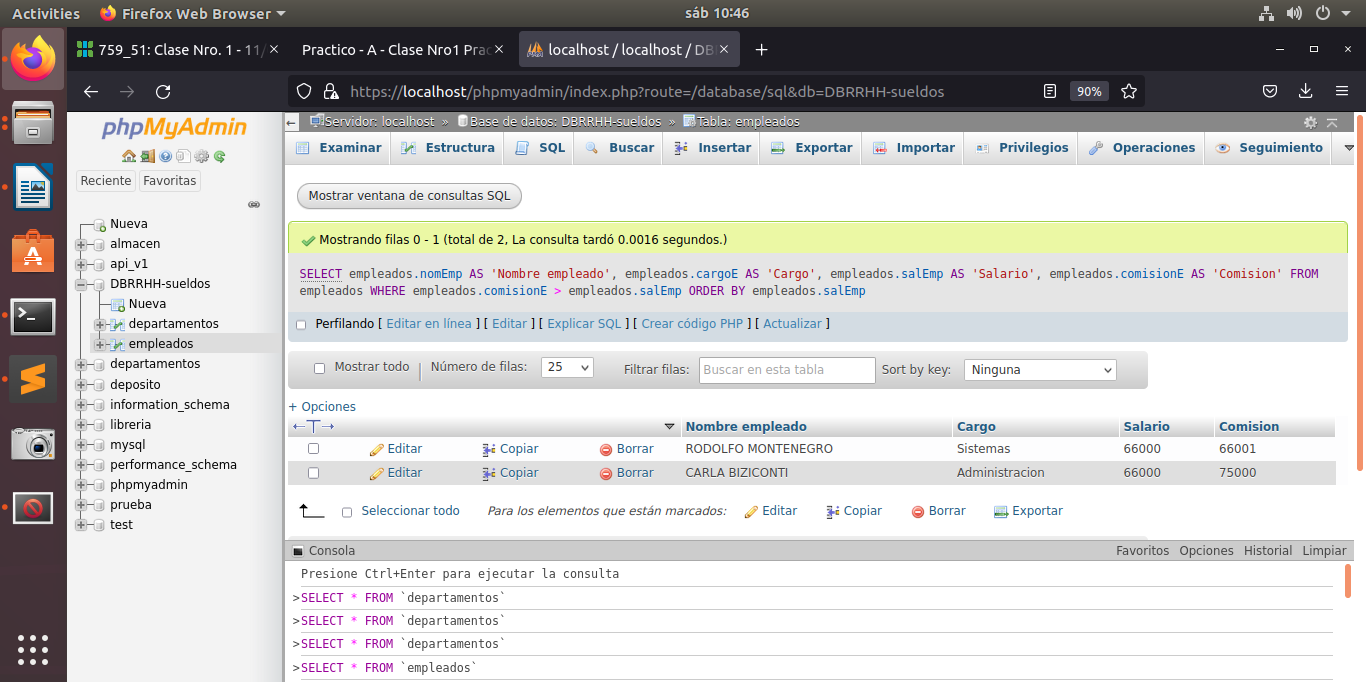
4.- Obtener la lista de los empleados que ganan una comisión superior a su sueldo.

Nota: *Como en la carga inicial se pusieron comisiones igual a mil (1000). Se haran cambios en dos empleados para que cumplan con esta condicion.*

SELECT empleados.nomEmp AS 'Nombre empleado', empleados.cargoE AS 'Cargo',

empleados.salEmp AS 'Salario', empleados.comisionE AS 'Comision' FROM empleados

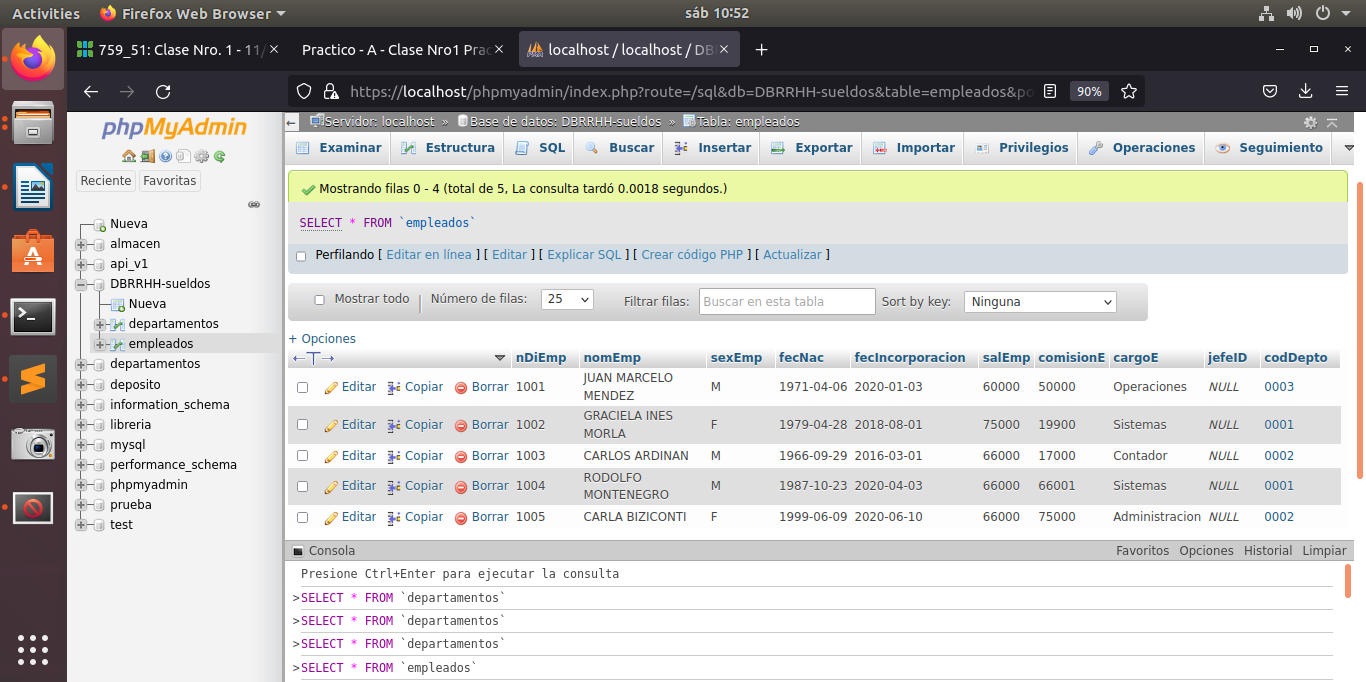
WHERE empleados.comisionE > empleados.salEmp ORDER BY empleados.salEmp



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

Nota: *Como en la carga inicial se pusieron comisiones igual a mil (1000). Se haran cambios en dos empleados para que cumplan con esta condicion.*

Consulta actual de empleados

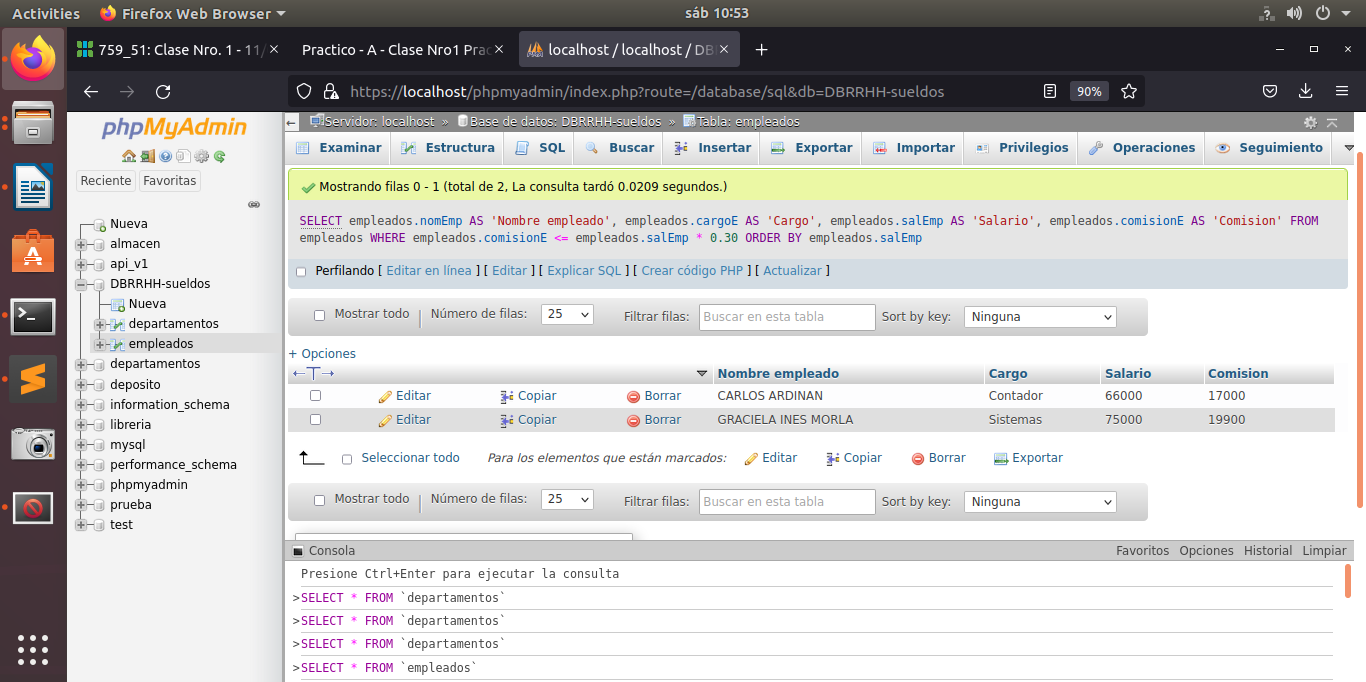


SELECT empleados.nomEmp AS 'Nombre empleado',

empleados.cargoE AS 'Cargo', empleados.salEmp AS 'Salario', empleados.comisionE AS 'Comision'

FROM empleados WHERE empleados.comisionE <= empleados.salEmp \* 0.30

ORDER BY empleados.salEmp



6. Muestra los empleados cuyo nombre empiece entre las letras J y Z (rango). Liste estos empleados

y su cargo por orden alfabético.

Nota: *se agregan empleados a efectos de mejorar el resultado.*

*Agregando empleados*

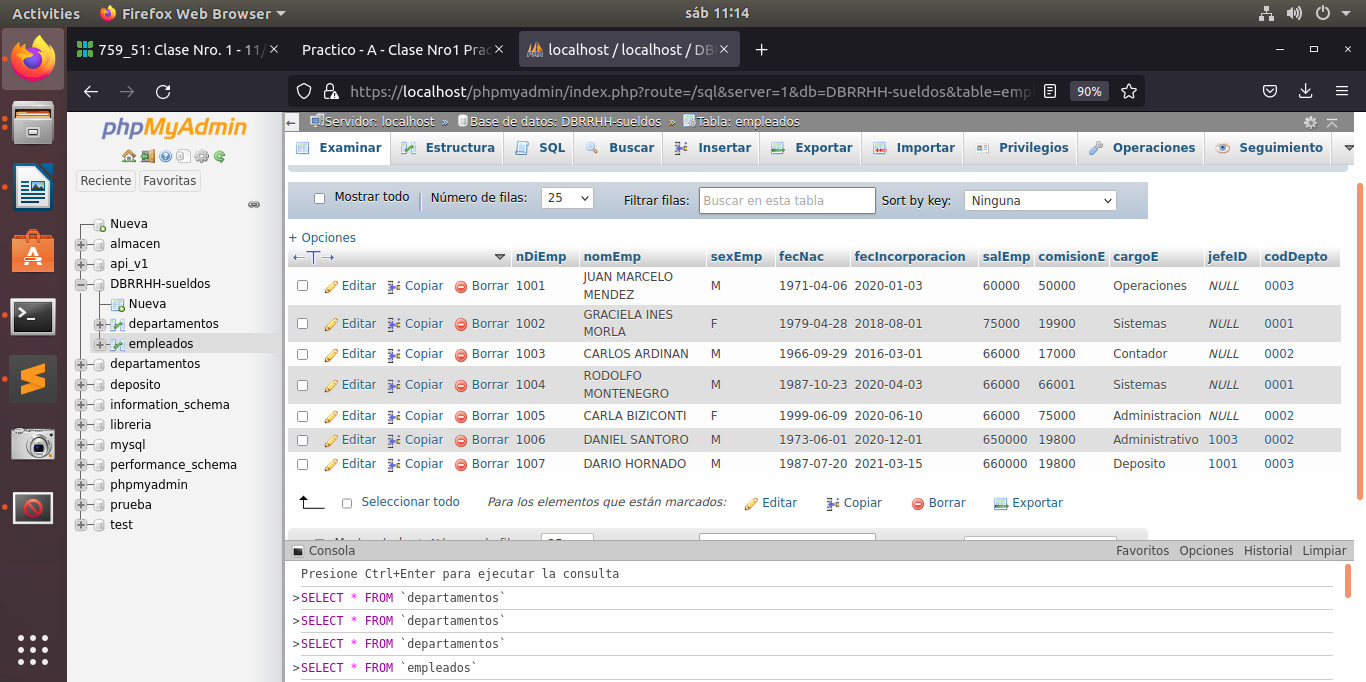
*INSERT INTO `empleados` (`nDiEmp`, `nomEmp`, `sexEmp`, `fecNac`, `fecIncorporacion`, `salEmp`, `comisionE`, `cargoE`, `jefeID`, `codDepto`)*

*VALUES ('1006', 'DANIEL SANTORO', 'M', '1973-06-01', '2020-12-01', '650000', '19800', 'Administrativo', '1003', '0002');*

*INSERT INTO `empleados` (`nDiEmp`, `nomEmp`, `sexEmp`, `fecNac`, `fecIncorporacion`, `salEmp`, `comisionE`, `cargoE`, `jefeID`, `codDepto`)*

*VALUES ('1007', 'DARIO HORNADO', 'M', '1987-07-20', '2021-03-15', '660000', '19800', 'Deposito', '1001', '0003');*

Consulta actual de empleados



SELECT empleados.nomEmp AS 'Nombre empleado', empleados.cargoE AS 'Cargo'

FROM empleados WHERE empleados.nomEmp >= 'J' AND empleados.nomEmp <= 'Z'

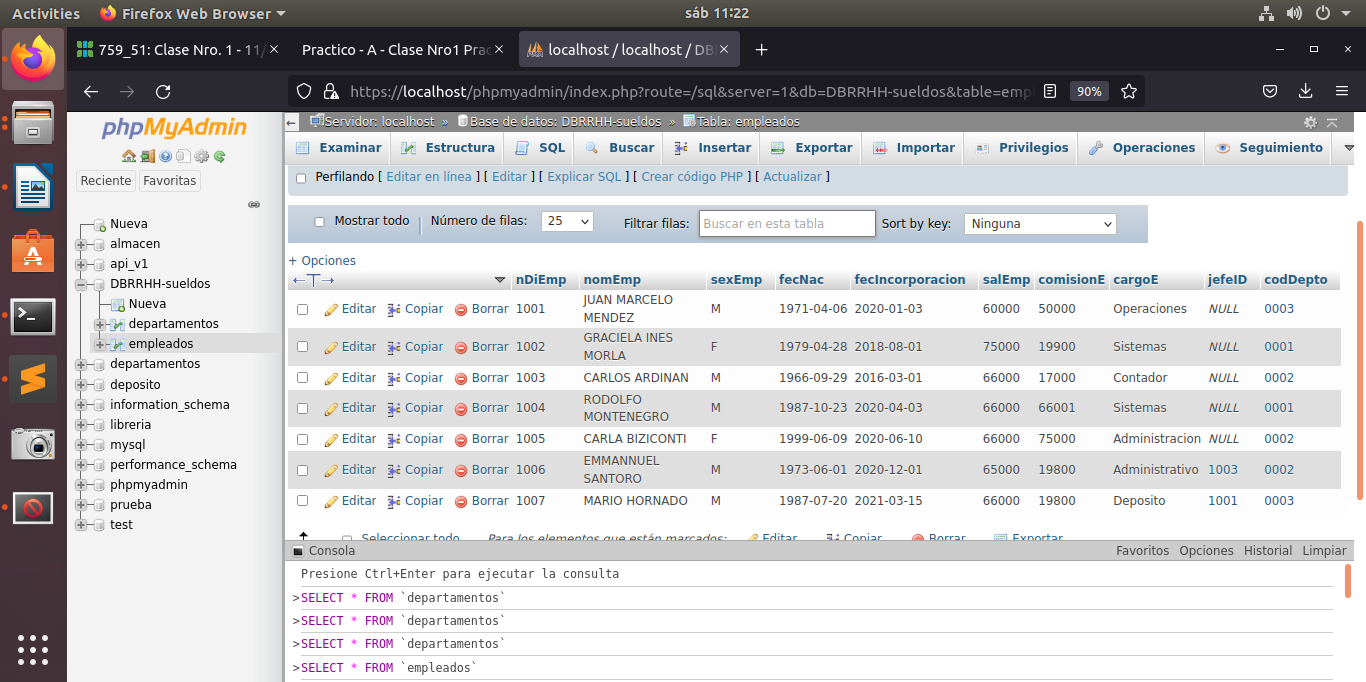
ORDER BY empleados.cargoE ASC



7. Hallar los empleados cuyo nombre no contiene la cadena “MA”

Nota: *se modifican empleados a efectos de mejorar el resultado.*

Consulta actual de empleados – modificandos a dos empleados el nombre por uno que no tenga la cadena “MA”

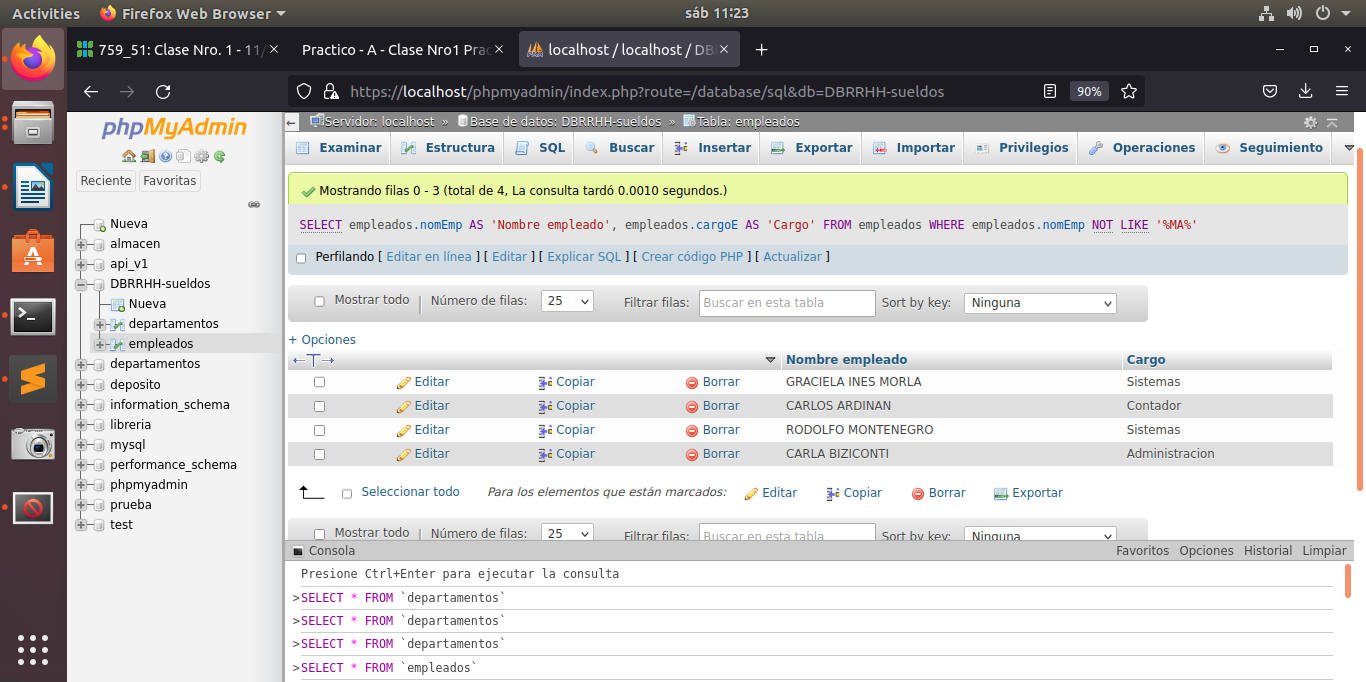


SELECT empleados.nomEmp AS 'Nombre empleado',

empleados.cargoE AS 'Cargo'

FROM empleados

WHERE empleados.nomEmp NOT LIKE '%MA%';



8. 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’

Nota: *se agregan empleados y departamento, a efectos de mejorar el resultado.*

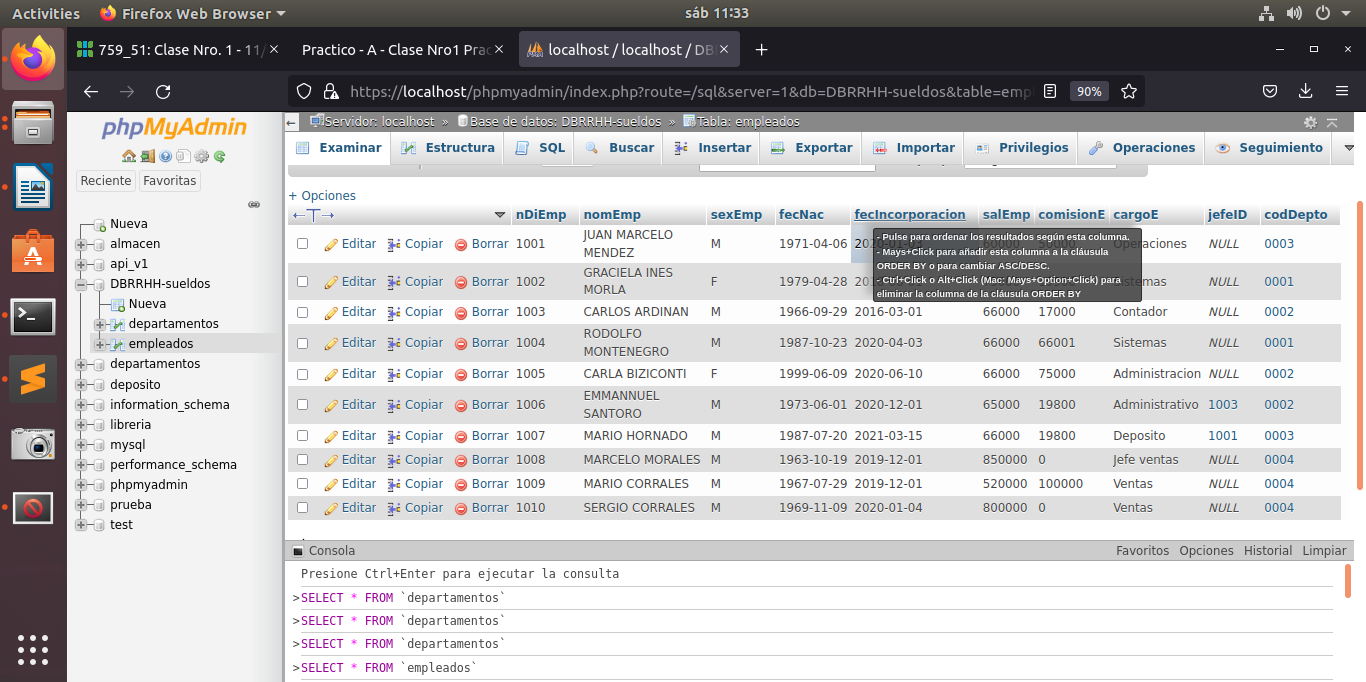
*INSERT INTO `departamentos` (`codDepto`, `nombreDpto`, `ciudad`, `codDirector`) VALUES ('0004', 'Ventas', 'CABA', NULL);*

*INSERT INTO `empleados` (`nDiEmp`, `nomEmp`, `sexEmp`, `fecNac`, `fecIncorporacion`, `salEmp`, `comisionE`, `cargoE`, `jefeID`, `codDepto`) VALUES ('1008', 'MARCELO MORALES', 'M', '1963-10-19', '2019-12-01', '850000', '0', 'Jefe ventas', NULL, '0004');*

*INSERT INTO `empleados` (`nDiEmp`, `nomEmp`, `sexEmp`, `fecNac`, `fecIncorporacion`, `salEmp`, `comisionE`, `cargoE`, `jefeID`, `codDepto`) VALUES ('1009', 'MARIO CORRALES', 'M', '1967-07-29', '2019-12-01', '520000', '100000', 'Ventas', NULL, '0004');*

*INSERT INTO `empleados` (`nDiEmp`, `nomEmp`, `sexEmp`, `fecNac`, `fecIncorporacion`, `salEmp`, `comisionE`, `cargoE`, `jefeID`, `codDepto`) VALUES ('1010', 'SERGIO CORRALES', 'M', '1969-11-09', '2020-01-04', '800000', '0', 'Ventas', NULL, '0004');*

*Consulta actual con empleados nuevo*

**

SELECT empleados.nDIEmp AS 'Codigo empleado', empleados.nomEmp AS 'Nombre empleado',

empleados.sexEmp AS 'Sexo', empleados.fecNac AS 'Fecha nacimiento',

empleados.fecIncorporacion AS 'Fecha incorpora.', empleados.salEmp AS 'Salario',

empleados.comisionE AS 'Comision', empleados.cargoE AS 'Cargo', empleados.jefeID AS 'codigo del Jefe',

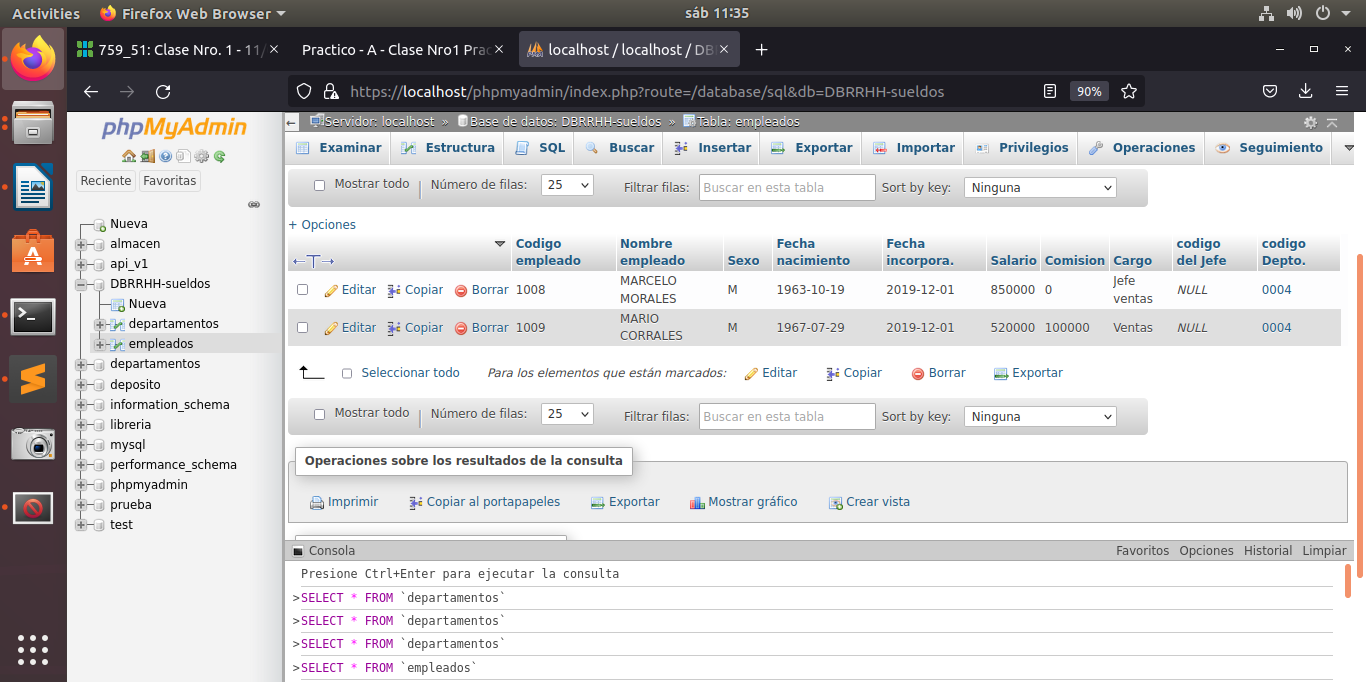
empleados.codDepto AS 'codigo Depto.'

FROM empleados

WHERE empleados.nomEmp LIKE 'M%'

AND empleados.salEmp > 800000 OR empleados.comisionE > 0

AND empleados.codDepto = 0004

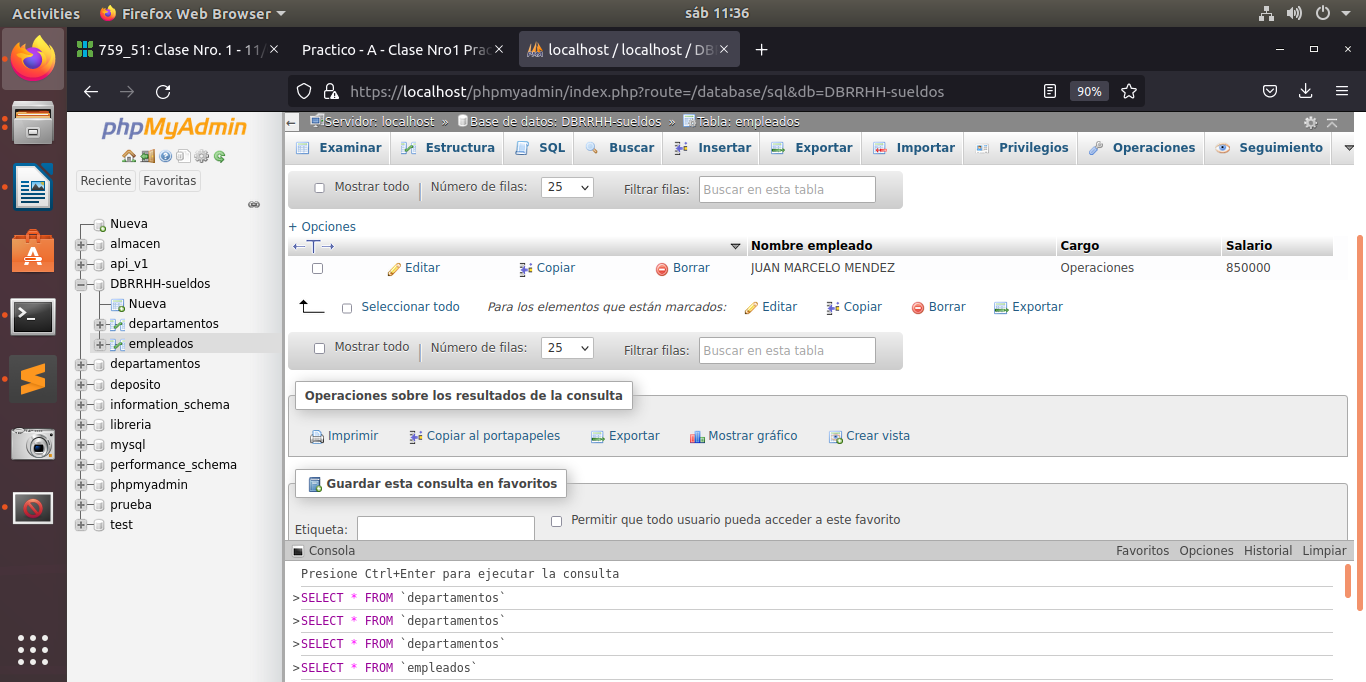


9. Mostrar el salario más alto de la empresa.

SELECT empleados.nomEmp AS 'Nombre empleado', empleados.cargoE AS 'Cargo',

MAX(empleados.salEmp) AS 'Salario'

FROM empleados

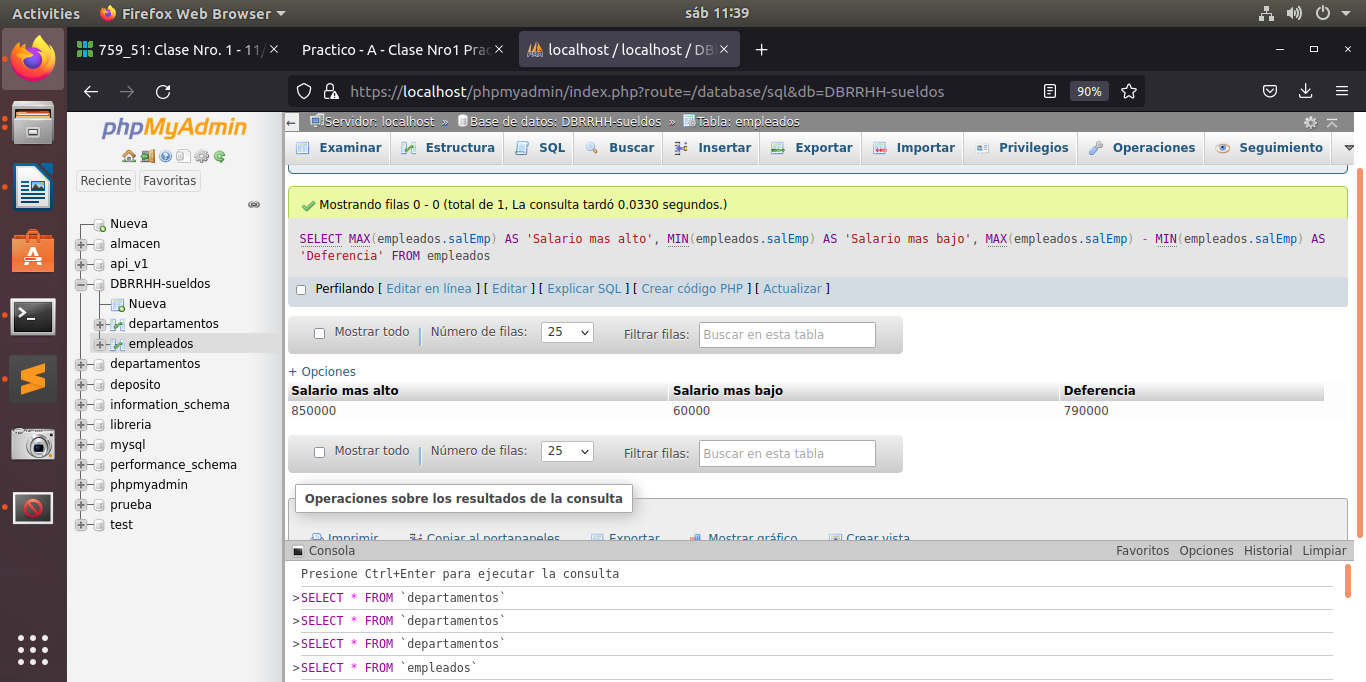


10. Hallar el salario más alto, el más bajo y la diferencia entre ellos.

SELECT MAX(empleados.salEmp) AS 'Salario mas alto', MIN(empleados.salEmp) AS 'Salario mas bajo',

MAX(empleados.salEmp) - MIN(empleados.salEmp) AS 'Deferencia'

FROM empleados



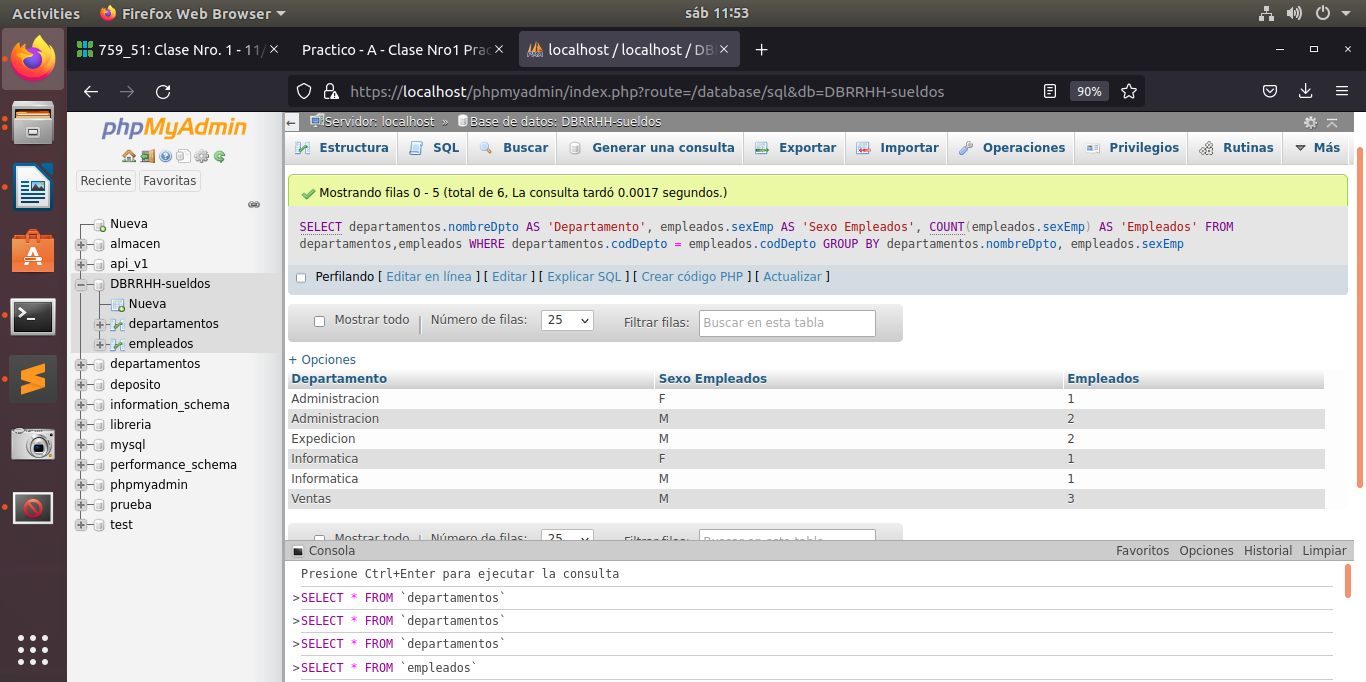
11. Mostrar el número de empleados de sexo femenino y de sexo masculino, por departamento.

SELECT departamentos.nombreDpto AS 'Departamento', empleados.sexEmp AS 'Sexo Empleados',

COUNT(empleados.sexEmp) AS 'Empleados' FROM departamentos,empleados

WHERE departamentos.codDepto = empleados.codDepto

GROUP BY departamentos.nombreDpto, empleados.sexEmp



12. Hallar los departamentos que tienen más de tres empleados. Mostrar el número de empleados de

esos departamentos.

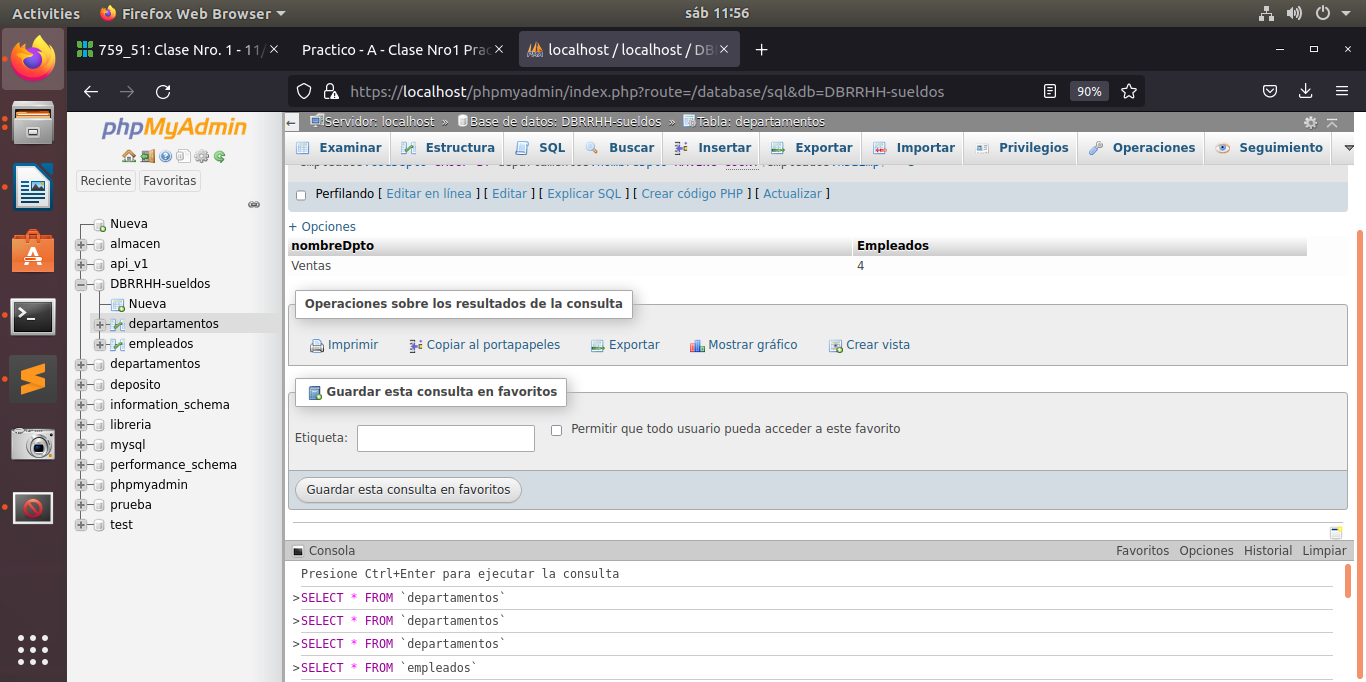
Nota: *se agrega empleado a efectos de mejorar el resultado.*

INSERT INTO `empleados` (`nDiEmp`, `nomEmp`, `sexEmp`, `fecNac`, `fecIncorporacion`, `salEmp`, `comisionE`, `cargoE`, `jefeID`, `codDepto`) VALUES ('1011', 'GERMAN LUX', 'M', '1990-04-24', '2018-03-01', '95000', '35000', 'Ventas', '1008', '0004');

SELECT departamentos.nombreDpto, COUNT(empleados.sexEmp) AS 'Empleados'

FROM departamentos,empleados WHERE departamentos.codDepto = empleados.codDepto

GROUP BY departamentos.nombreDpto HAVING COUNT(empleados.nDiEmp) > 3



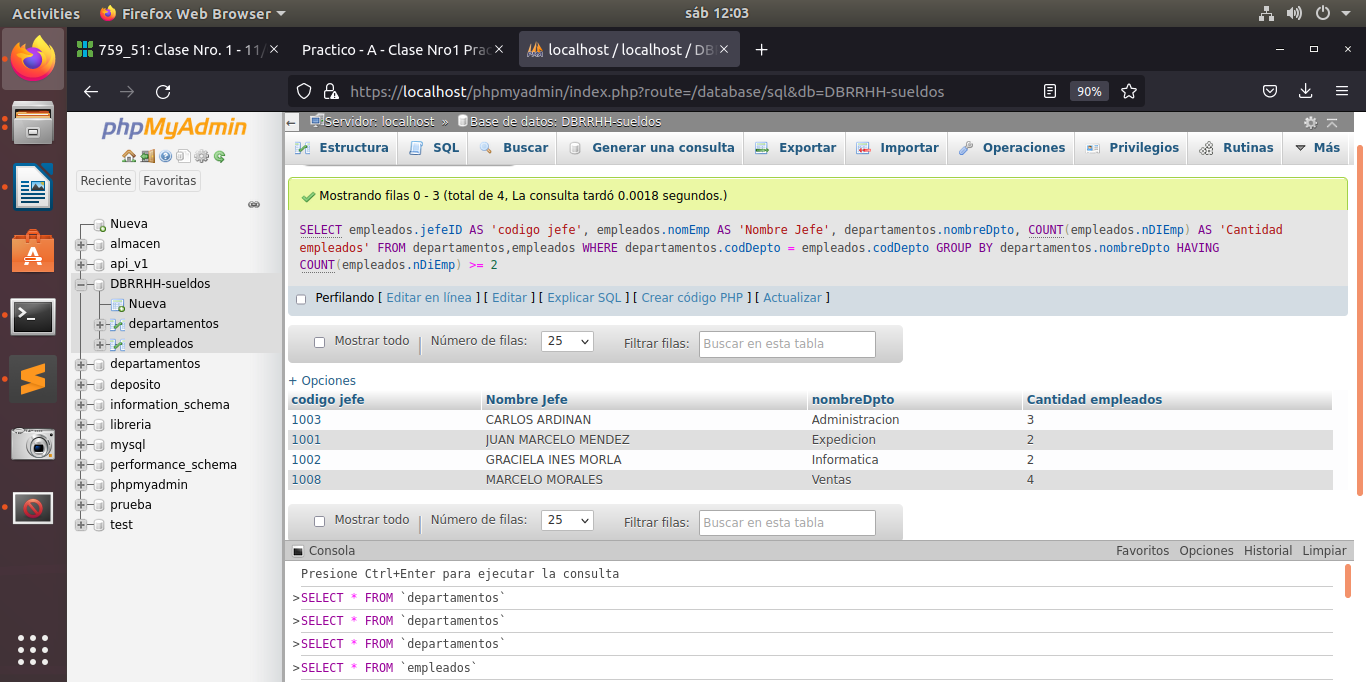
13. Mostrar el código y nombre de cada jefe, junto al número de empleados que dirige. Solo los que

tengan más de dos empleados (2 incluido).

SELECT empleados.jefeID AS 'codigo jefe', empleados.nomEmp AS 'Nombre Jefe', departamentos.nombreDpto, COUNT(empleados.nDIEmp) AS 'Cantidad empleados'

FROM departamentos,empleados WHERE departamentos.codDepto = empleados.codDepto

GROUP BY departamentos.nombreDpto HAVING COUNT(empleados.nDiEmp) >= 2



14. Hallar los departamentos que no tienen empleados

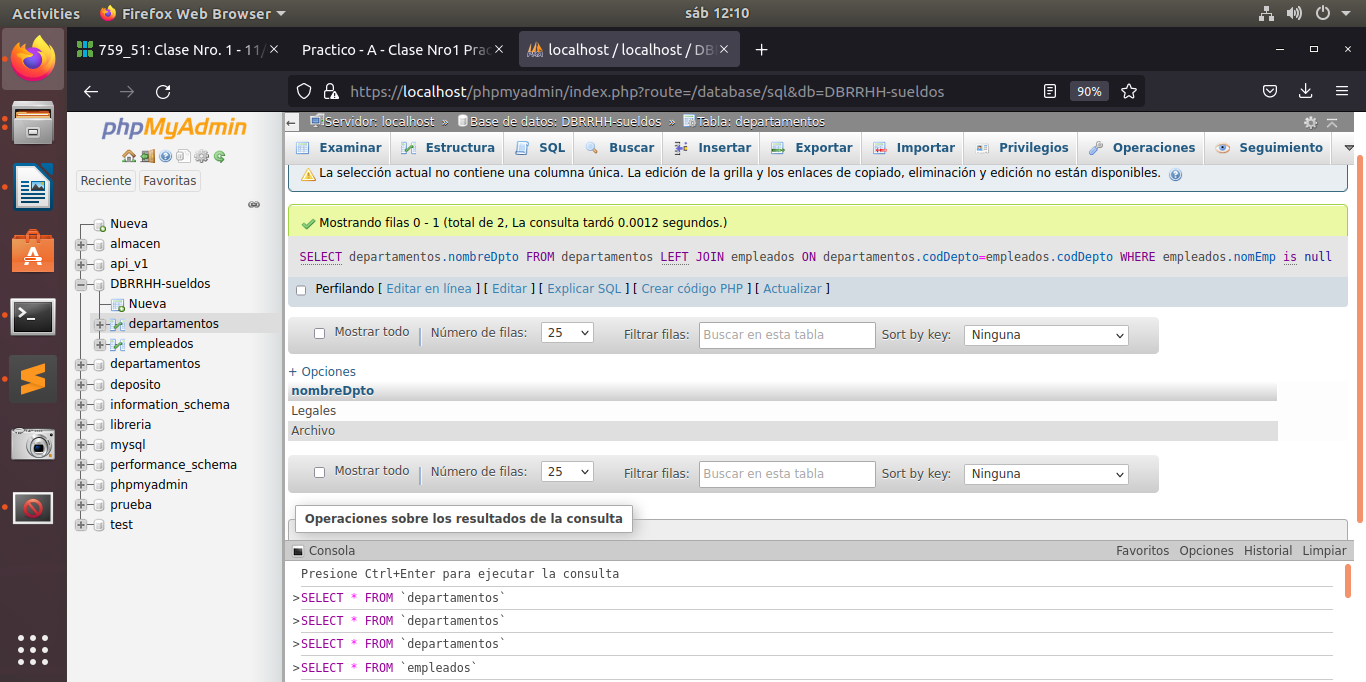
*Nota: se agrega un departamento a efectos de mejorar el resultado.*

*INSERT INTO `departamentos` (`codDepto`, `nombreDpto`, `ciudad`, `codDirector`) VALUES ('0005', 'Legales', 'BsAs', NULL), ('0006', 'Archivo', 'CABA', NULL);*

*SELECT departamentos.nombreDpto FROM departamentos*

*LEFT JOIN empleados ON departamentos.codDepto=empleados.codDepto*

*WHERE empleados.nomEmp is null*



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

*la suma.*

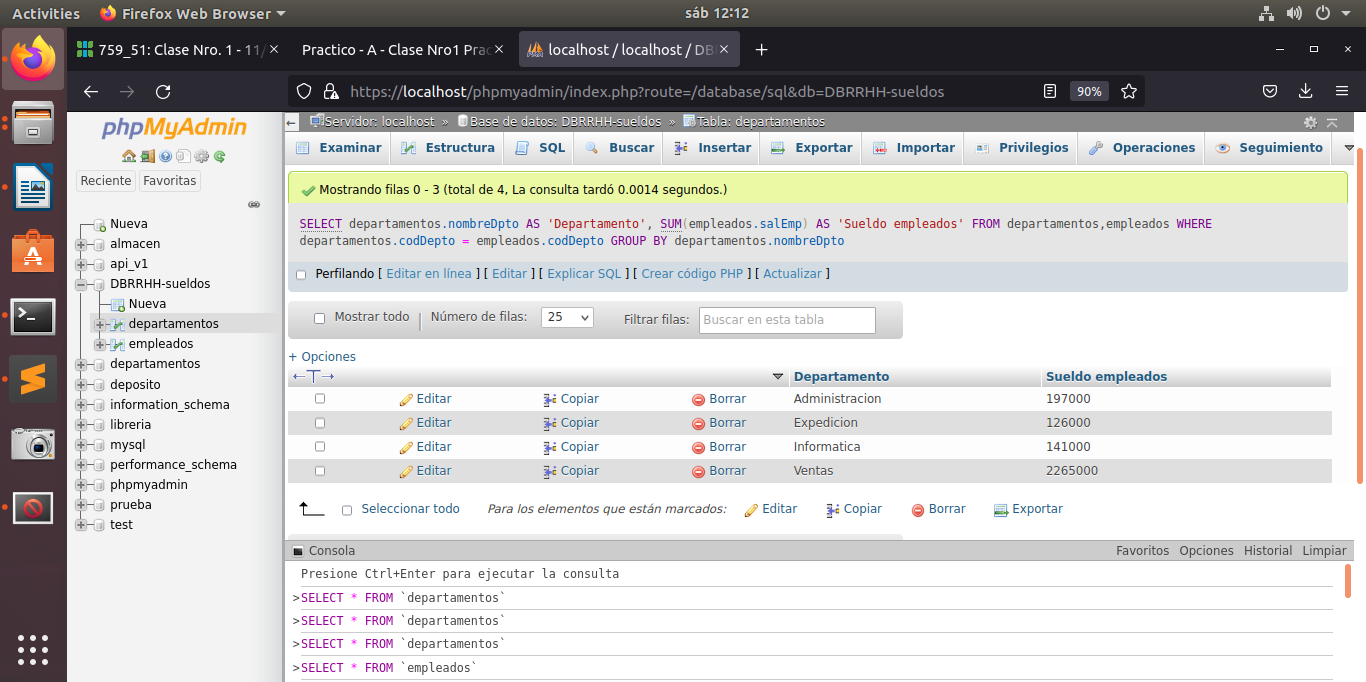
*SELECT departamentos.nombreDpto AS 'Departamento',*

*SUM(empleados.salEmp) AS 'Sueldo empleados'*

*FROM departamentos,empleados*

*WHERE departamentos.codDepto = empleados.codDepto*

*GROUP BY departamentos.nombreDpto*

**