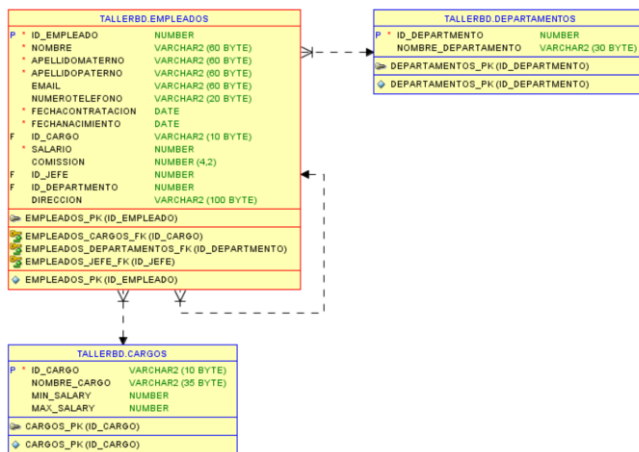


## Consulta



### 1. Abrir shell

python manage.py shell

```

PS C:\workspacedjango\ejemploBD> python manage.py shell
Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
  
```

### 2. cargar el modelo

from rrhh.models import Cargo, Departamento, Empleado

```

>>> from rrhh.models import Cargo, Departamento, Empleado
>>>
  
```

### 3. Consultar por todos los empleados

```

>>> qs=Empleado.objects.all()
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`, `rrhh_empleado`.`nombre`, `rrhh_empleado`.`apellidoPaterno`, `rrhh_empleado`.`apellidoMaterno`, `rrhh_empleado`.`email`, `rrhh_empleado`.`telefono`, `rrhh_empleado`.`direccion`, `rrhh_empleado`.`fechaContratacion`, `rrhh_empleado`.`fechaNacimiento`, `rrhh_empleado`.`salario`, `rrhh_empleado`.`comision`, `rrhh_empleado`.`cargoId_id`, `rrhh_empleado`.`departamentoId_id` FROM `rrhh_empleado`'
>>>
  
```

```

>>> empleado = Empleado.objects.all()
>>> str(empleado.query)
'SELECT `rrhh_empleado`.`idEmpleado`, `rrhh_empleado`.`nombre`, `rrhh_empleado`.`apellidoPaterno`, `rrhh_empleado`.`apellidoMaterno`, `rrhh_empleado`.`email`, `rrhh_empleado`.`telefono`, `rrhh_empleado`.`direccion`, `rrhh_empleado`.`fechaContratacion`, `rrhh_empleado`.`fechaNacimiento`, `rrhh_empleado`.`salario`, `rrhh_empleado`.`comision`, `rrhh_empleado`.`cargoId_id`, `rrhh_empleado`.`departamentoId_id` FROM `rrhh_empleado`'
>>>
  
```

4. Consultar cantidad de todos los empleados (método count() que no retorna conjuntos de consulta)

```
>>> cantidadEmpleados = Empleado.objects.all().count()
>>> cantidadEmpleados
107
```

5. Si requiere usar funciones de agregaciones como: Avg, Count, Max, Min, Sum, Variance, StdDev

- Cargar las funciones de base de datos

```
>>> from django.db.models import Avg, Max, Min, Sum, Count
>>>
```

- Obtener el promedio de salario de todos los empleados

```
>>> promedio = Empleado.objects.all().aggregate(Avg('salario'))
>>> promedio
{'salario__avg': 6461.6822}
>>>
```

- Obtener el salario máximo de todos los empleados

```
>>> maxSalario = Empleado.objects.all().aggregate(Max('salario'))
>>> maxSalario
{'salario__max': 24000}
```

- Obtener el salario mínimo de todos los empleados

```
>>> minSalario = Empleado.objects.all().aggregate(Min('salario'))
>>> minSalario
{'salario__min': 2100}
```

- Obtener la suma de todos los salarios de todos los empleados

```
>>> sumaTotalSalario = Empleado.objects.all().aggregate(Sum('salario'))
>>> sumaTotalSalario
{'salario__sum': 691400}
>>>
```

- Obtener la cantidad de todos los empleados

```
>>> cantidadEmpleados = Empleado.objects.all().aggregate(Count('idEmpleado'))
>>> cantidadEmpleados
{'idEmpleado__count': 107}
```

Filtrar una tabla con una o más condiciones

6. Obtener los empleados del departamento de transporte id=50

```
>>> qs=Empleado.objects.filter(departamentoId=50)
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`, `rrhh_empleado`.`nombre`, `rrhh_empleado`.`apellidoPaterno`, `rrhh_empleado`.`apellidoMaterno`, `rrhh_empleado`.`email`, `rrhh_empleado`.`telefono`, `rrhh_empleado`.`direccion`, `rrhh_empleado`.`fechaContratacion`, `rrhh_empleado`.`fechaNacimiento`, `rrhh_empleado`.`salario`, `rrhh_empleado`.`comision`, `rrhh_empleado`.`cargoId_id`, `rrhh_empleado`.`departamentoId_id` FROM `rrhh_empleado` WHERE `rrhh_empleado`.`departamentoId_id` = 50'
>>>
>>>
```

```
>>> qs=Empleado.objects.filter(departamentoId=50)
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`,
`rrhh_empleado`.`nombre`,
`rrhh_empleado`.`apellidoPaterno`,
`rrhh_empleado`.`apellidoMaterno`,
`rrhh_empleado`.`email`,
`rrhh_empleado`.`telefono`,
`rrhh_empleado`.`direccion`,
`rrhh_empleado`.`fechaContratacion`,
`rrhh_empleado`.`fechaNacimiento`,
`rrhh_empleado`.`salario`,
`rrhh_empleado`.`comision`,
`rrhh_empleado`.`cargoId_id`,
`rrhh_empleado`.`departamentoId_id`
FROM `rrhh_empleado`
WHERE `rrhh_empleado`.`departamentoId_id` = 50'
>>>
```

La búsqueda posee 45 registros

```
>>> qs=Empleado.objects.filter(departamentoId=50).count()
>>> qs
45
```

7. Obtener los empleados del departamento de transporte id=50 y cargoid= SH\_CLERK

```
>>> qs=Empleado.objects.filter(departamentoId=50,cargoId='SH_CLERK')
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`, `rrhh_empleado`.`nombre`, `rrhh_empleado`.`apellidoPaterno`, `rrhh_empleado`.`apellidoMaterno`, `rrhh_empleado`.`email`, `rrhh_empleado`.`telefono`, `rrhh_empleado`.`direccion`, `rrhh_empleado`.`fechaContratacion`, `rrhh_empleado`.`fechaNacimiento`, `rrhh_empleado`.`salario`, `rrhh_empleado`.`comision`, `rrhh_empleado`.`cargoId_id`, `rrhh_empleado`.`departamentoId_id` FROM `rrhh_empleado` WHERE (`rrhh_empleado`.`cargoId_id` = SH_CLERK AND `rrhh_empleado`.`departamentoId_id` = 50)'
>>>
```

```
>>> qs=Empleado.objects.filter(departamentoId=50,cargoId='SH_CLERK')
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`,
`rrhh_empleado`.`nombre`,
`rrhh_empleado`.`apellidoPaterno`,
`rrhh_empleado`.`apellidoMaterno`,
`rrhh_empleado`.`email`,
`rrhh_empleado`.`telefono`,
`rrhh_empleado`.`direccion`,
`rrhh_empleado`.`fechaContratacion`,
`rrhh_empleado`.`fechaNacimiento`,
`rrhh_empleado`.`salario`,
`rrhh_empleado`.`comision`,
`rrhh_empleado`.`cargoId_id`,
`rrhh_empleado`.`departamentoId_id`
FROM `rrhh_empleado`
WHERE (`rrhh_empleado`.`cargoId_id` = SH_CLERK AND `rrhh_empleado`.`departamentoId_id` = 50)'
>>>
```

Cantidad de registros obtenidos es 20

```
>>>
>>> qs=Empleado.objects.filter(departamentoId=50,cargoId='SH_CLERK').count()
>>> qs
20
```

Obtener los empleados de nombre Steven

```
>>> qs=Empleado.objects.filter(nombre='Steven')
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`, `rrhh_empleado`.`nombre`, `rrhh_empleado`.`apellidoPaterno`, `rrhh_empleado`.`apellidoMaterno`, `rrhh_empleado`.`email`, `rrhh_empleado`.`telefono`, `rrhh_empleado`.`direccion`, `rrhh_empleado`.`fechaContratacion`, `rrhh_empleado`.`fechaNacimiento`, `rrhh_empleado`.`salario`, `rrhh_empleado`.`comision`, `rrhh_empleado`.`cargoId_id`, `rrhh_empleado`.`departamentoId_id` FROM `rrhh_empleado` WHERE `rrhh_empleado`.`nombre` = Steven'
>>>
```

```
>>> qs=Empleado.objects.filter(nombre='Steven')
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`,
`rrhh_empleado`.`nombre`,
`rrhh_empleado`.`apellidoPaterno`,
`rrhh_empleado`.`apellidoMaterno`,
`rrhh_empleado`.`email`,
`rrhh_empleado`.`telefono`,
`rrhh_empleado`.`direccion`,
`rrhh_empleado`.`fechaContratacion`,
`rrhh_empleado`.`fechaNacimiento`,
`rrhh_empleado`.`salario`,
`rrhh_empleado`.`comision`,
`rrhh_empleado`.`cargoId_id`,
`rrhh_empleado`.`departamentoId_id`
FROM `rrhh_empleado`
WHERE `rrhh_empleado`.`nombre` = Steven'
```

Obtener los empleados cuyo nombre comience con Di

```
>>> qs=Empleado.objects.filter(nombre__startswith='Di')
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`, `rrhh_empleado`.`nombre`, `rrhh_empleado`.`apellidoPaterno`, `rrhh_empleado`.`apellidoMaterno`, `rrhh_empleado`.`email`, `rrhh_empleado`.`telefono`, `rrhh_empleado`.`direccion`, `rrhh_empleado`.`fechaContratacion`, `rrhh_empleado`.`fechaNacimiento`, `rrhh_empleado`.`salario`, `rrhh_empleado`.`comision`, `rrhh_empleado`.`cargoId_id`, `rrhh_empleado`.`departamentoId_id` FROM `rrhh_empleado` WHERE `rrhh_empleado`.`nombre` LIKE BINARY Di%'
>>>
```

```
>>> qs=Empleado.objects.filter(nombre__startswith='Di')
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`,
`rrhh_empleado`.`nombre`,
`rrhh_empleado`.`apellidoPaterno`,
`rrhh_empleado`.`apellidoMaterno`,
`rrhh_empleado`.`email`,
`rrhh_empleado`.`telefono`,
`rrhh_empleado`.`direccion`,
`rrhh_empleado`.`fechaContratacion`,
`rrhh_empleado`.`fechaNacimiento`,
`rrhh_empleado`.`salario`,
`rrhh_empleado`.`comision`,
`rrhh_empleado`.`cargoId_id`,
`rrhh_empleado`.`departamentoId_id`
FROM `rrhh_empleado`
WHERE `rrhh_empleado`.`nombre` LIKE BINARY Di%'
```

Obtener empleados cuyo nombre contenga "lli"

```
>>> qs=Empleado.objects.filter(nombre__contains='lli')
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`, `rrhh_empleado`.`nombre`, `rrhh_empleado`.`apellidoPaterno`, `rrhh_empleado`.`apellidoMaterno`, `rrhh_empleado`.`email`, `rrhh_empleado`.`telefono`, `rrhh_empleado`.`direccion`, `rrhh_empleado`.`fechaContratacion`, `rrhh_empleado`.`fechaNacimiento`, `rrhh_empleado`.`salario`, `rrhh_empleado`.`comision`, `rrhh_empleado`.`cargoId_id`, `rrhh_empleado`.`departamentoId_id` FROM `rrhh_empleado` WHERE `rrhh_empleado`.`nombre` LIKE BINARY %lli%'
>>>
```

```
>>> qs=Empleado.objects.filter(nombre__contains='lli')
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`,
`rrhh_empleado`.`nombre`,
`rrhh_empleado`.`apellidoPaterno`,
`rrhh_empleado`.`apellidoMaterno`,
`rrhh_empleado`.`email`,
`rrhh_empleado`.`telefono`,
`rrhh_empleado`.`direccion`,
`rrhh_empleado`.`fechaContratacion`,
`rrhh_empleado`.`fechaNacimiento`,
`rrhh_empleado`.`salario`,
`rrhh_empleado`.`comision`,
`rrhh_empleado`.`cargoId_id`,
`rrhh_empleado`.`departamentoId_id`
FROM `rrhh_empleado`
WHERE `rrhh_empleado`.`nombre` LIKE BINARY %lli%'
```

-----  
distinct('field\_name') is not supported in MySQL. It only support distinct(). distinct('field\_name') will only work on **PostgreSQL**. For more details, please check the [documentation](#).

```
>>> qs=Empleado.objects.values('departamentoId').distinct()
>>> str(qs.query)
'SELECT DISTINCT `rrhh_empleado`.`departamentoId_id` FROM `rrhh_empleado` '
>>>
```

```
>>> qs=Empleado.objects.values('departamentoId').distinct()
>>> str(qs.query)
'SELECT DISTINCT `rrhh_empleado`.`departamentoId_id`
FROM `rrhh_empleado` '
```

```
>>> cantidadDepartamentos=Empleado.objects.values('departamentoId').distinct().count()
>>> cantidadDepartamentos
11
```

1. (Select de dos tablas) Obtener toda la información de los empleados. Indicar también el nombre del cargo que posee el empleado

```
>>> empleado = Empleado.objects.all().select_related("cargoId")
>>> empleado
<QuerySet [(<Empleado: Empleado object (100)>)]>
>>> str(empleado.query)
'SELECT `rrhh_empleado`.`idEmpleado`, `rrhh_empleado`.`nombre`, `rrhh_empleado`.`apellidoPaterno`, `rrhh_empleado`.`apellidoMaterno`, `rrhh_empleado`.`email`, `rrhh_empleado`.`telefono`, `rrhh_empleado`.`direccion`, `rrhh_empleado`.`fechaContratacion`, `rrhh_empleado`.`fechaNacimiento`, `rrhh_empleado`.`salario`, `rrhh_empleado`.`comision`, `rrhh_empleado`.`cargoId_id`, `rrhh_empleado`.`departamentoId_id`, `rrhh_cargo`.`cargoId`, `rrhh_cargo`.`cargoNombre`, `rrhh_cargo`.`minSalario`, `rrhh_cargo`.`maxSalario` FROM `rrhh_empleado` LEFT OUTER JOIN `rrhh_cargo` ON (`rrhh_empleado`.`cargoId_id` = `rrhh_cargo`.`cargoId`)'
>>> █
```

```
>>> empleado = Empleado.objects.all().select_related("cargoId")
>>> empleado
<QuerySet [(<Empleado: Empleado object (100)>)]>
>>> str(empleado.query)
>>> str(empleado.query)
'SELECT `rrhh_empleado`.`idEmpleado`,
`rrhh_empleado`.`nombre`,
`rrhh_empleado`.`apellidoPaterno`,
`rrhh_empleado`.`apellidoMaterno`,
`rrhh_empleado`.`email`,
`rrhh_empleado`.`telefono`,
`rrhh_empleado`.`direccion`,
`rrhh_empleado`.`fechaContratacion`,
`rrhh_empleado`.`fechaNacimiento`,
`rrhh_empleado`.`salario`,
`rrhh_empleado`.`comision`,
`rrhh_empleado`.`cargoId_id`,
`rrhh_empleado`.`departamentoId_id`,
`rrhh_cargo`.`cargoId`,
`rrhh_cargo`.`cargoNombre`,
`rrhh_cargo`.`minSalario`,
`rrhh_cargo`.`maxSalario`
FROM `rrhh_empleado` LEFT OUTER JOIN `rrhh_cargo`
ON (`rrhh_empleado`.`cargoId_id` = `rrhh_cargo`.`cargoId`)'
>>> █
```

2. (Select de tres tablas) Obtener la información de los empleados. Indicar también el nombre del cargo que posee el empleado y el nombre del departamento en la que pertenecen.

```
>>> empleado = Empleado.objects.all().select_related("cargoId", "departamentoId")
>>> str(empleado.query)
'SELECT `rrhh_empleado`.`idEmpleado`, `rrhh_empleado`.`nombre`, `rrhh_empleado`.`apellidoPaterno`, `rrhh_empleado`.`apellidoMaterno`, `rrhh_empleado`.`email`, `rrhh_empleado`.`telefono`, `rrhh_empleado`.`direccion`, `rrhh_empleado`.`fechaContratacion`, `rrhh_empleado`.`fechaNacimiento`, `rrhh_empleado`.`salario`, `rrhh_empleado`.`comision`, `rrhh_empleado`.`cargoId_id`, `rrhh_empleado`.`departamentoId_id`, `rrhh_cargo`.`cargoId`, `rrhh_cargo`.`cargoNombre`, `rrhh_cargo`.`minSalario`, `rrhh_cargo`.`maxSalario`, `rrhh_departamento`.`departId`, `rrhh_departamento`.`departNombre` FROM `rrhh_empleado` LEFT OUTER JOIN `rrhh_cargo` ON (`rrhh_empleado`.`cargoId_id` = `rrhh_cargo`.`cargoId`) LEFT OUTER JOIN `rrhh_departamento` ON (`rrhh_empleado`.`departamentoId_id` = `rrhh_departamento`.`departId`)'
>>> █
```

```
>>> empleado = Empleado.objects.all().select_related("cargoId", "departamentoId")
>>> str(empleado.query)
'SELECT
`rrhh_empleado`.`idEmpleado`,
`rrhh_empleado`.`nombre`,
`rrhh_empleado`.`apellidoPaterno`,
`rrhh_empleado`.`apellidoMaterno`,
`rrhh_empleado`.`email`,
`rrhh_empleado`.`telefono`,
`rrhh_empleado`.`direccion`,
`rrhh_empleado`.`fechaContratacion`,
`rrhh_empleado`.`fechaNacimiento`,
`rrhh_empleado`.`salario`,
`rrhh_empleado`.`comision`,
`rrhh_empleado`.`cargoId_id`,
`rrhh_empleado`.`departamentoId_id`,
`rrhh_cargo`.`cargoId`,
`rrhh_cargo`.`cargoNombre`,
`rrhh_cargo`.`minSalario`,
`rrhh_cargo`.`maxSalario`,
`rrhh_departamento`.`departId`,
`rrhh_departamento`.`departNombre`
FROM `rrhh_empleado` LEFT OUTER JOIN `rrhh_cargo` ON (`rrhh_empleado`.`cargoId_id` = `rrhh_cargo`.`cargoId`)
LEFT OUTER JOIN `rrhh_departamento` ON (`rrhh_empleado`.`departamentoId_id` = `rrhh_departamento`.`departId`)'
>>> █
```

3. (Select de una tabla con filtro) Obtener la información de los empleados que posean el cargo cuyo ID =AD\_PRES



```
>>> qs=Empleado.objects.filter(cargoId='AD_PRES')
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`, `rrhh_empleado`.`nombre`, `rrhh_empleado`.`apellidoPaterno`, `rrhh_empleado`.`apellidoMaterno`, `rrhh_empleado`.`email`, `rrhh_empleado`.`telefono`, `rrhh_empleado`.`direccion`, `rrhh_empleado`.`fechaContratacion`, `rrhh_empleado`.`fechaNacimiento`, `rrhh_empleado`.`salario`, `rrhh_empleado`.`comision`, `rrhh_empleado`.`cargoId_id`, `rrhh_empleado`.`departamentoId_id` FROM `rrhh_empleado` WHERE `rrhh_empleado`.`cargoId_id` = AD_PRES'
```

```
>>> qs=Empleado.objects.filter(cargoId='AD_PRES')
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`,
`rrhh_empleado`.`nombre`,
`rrhh_empleado`.`apellidoPaterno`,
`rrhh_empleado`.`apellidoMaterno`,
`rrhh_empleado`.`email`,
`rrhh_empleado`.`telefono`,
`rrhh_empleado`.`direccion`,
`rrhh_empleado`.`fechaContratacion`,
`rrhh_empleado`.`fechaNacimiento`,
`rrhh_empleado`.`salario`,
`rrhh_empleado`.`comision`,
`rrhh_empleado`.`cargoId_id`,
`rrhh_empleado`.`departamentoId_id`
FROM `rrhh_empleado`
WHERE
`rrhh_empleado`.`cargoId_id` = AD_PRES'
```

4. (Select de dos tablas con filtro) Obtener la información de los empleados que posean el cargo cuyo ID =AD\_PRES y que muestre el nombre del cargo con el min salario y máximo salario

```
>>> qs=Empleado.objects.select_related("cargoId").filter(cargoId__cargoId='AD_PRES')
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`, `rrhh_empleado`.`nombre`, `rrhh_empleado`.`apellidoPaterno`, `rrhh_empleado`.`apellidoMaterno`, `rrhh_empleado`.`email`, `rrhh_empleado`.`telefono`, `rrhh_empleado`.`direccion`, `rrhh_empleado`.`fechaContratacion`, `rrhh_empleado`.`fechaNacimiento`, `rrhh_empleado`.`salario`, `rrhh_empleado`.`comision`, `rrhh_empleado`.`cargoId_id`, `rrhh_empleado`.`departamentoId_id`, `rrhh_cargo`.`cargoId`, `rrhh_cargo`.`cargoNombre`, `rrhh_cargo`.`minSalario`, `rrhh_cargo`.`maxSalario` FROM `rrhh_empleado` INNER JOIN `rrhh_cargo` ON (`rrhh_empleado`.`cargoId_id` = `rrhh_cargo`.`cargoId`) WHERE `rrhh_empleado`.`cargoId_id` = AD_PRES'
```

```
>>> qs=Empleado.objects.select_related("cargoId").filter(cargoId__cargoId='AD_PRES')
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`,
`rrhh_empleado`.`nombre`,
`rrhh_empleado`.`apellidoPaterno`,
`rrhh_empleado`.`apellidoMaterno`,
`rrhh_empleado`.`email`,
`rrhh_empleado`.`telefono`,
`rrhh_empleado`.`direccion`,
`rrhh_empleado`.`fechaContratacion`,
`rrhh_empleado`.`fechaNacimiento`,
`rrhh_empleado`.`salario`,
`rrhh_empleado`.`comision`,
`rrhh_empleado`.`cargoId_id`,
`rrhh_empleado`.`departamentoId_id`,
`rrhh_cargo`.`cargoId`,
`rrhh_cargo`.`cargoNombre`,
`rrhh_cargo`.`minSalario`,
`rrhh_cargo`.`maxSalario`
FROM `rrhh_empleado` INNER JOIN `rrhh_cargo`
ON (`rrhh_empleado`.`cargoId_id` = `rrhh_cargo`.`cargoId`)
WHERE `rrhh_empleado`.`cargoId_id` = AD_PRES'
```

Esta malo

```
>>> from rrhh.models import Cargo, Departamento, Empleado
>>> qs=Empleado.objects.filter(cargoId__cargoId='AD_PRES')
>>> str(qs.query)
'SELECT `rrhh_empleado`.`idEmpleado`, `rrhh_empleado`.`nombre`, `rrhh_empleado`.`apellidoPaterno`, `rrhh_empleado`.`apellidoMaterno`, `rrhh_empleado`.`email`, `rrhh_empleado`.`telefono`, `rrhh_empleado`.`direccion`, `rrhh_empleado`.`fechaContratacion`, `rrhh_empleado`.`fechaNacimiento`, `rrhh_empleado`.`salario`, `rrhh_empleado`.`comision`, `rrhh_empleado`.`cargoId_id`, `rrhh_empleado`.`departamentoId_id` FROM `rrhh_empleado` WHERE `rrhh_empleado`.`cargoId_id` = AD_PRES'
>>>
```