

# Tarea para BD04.

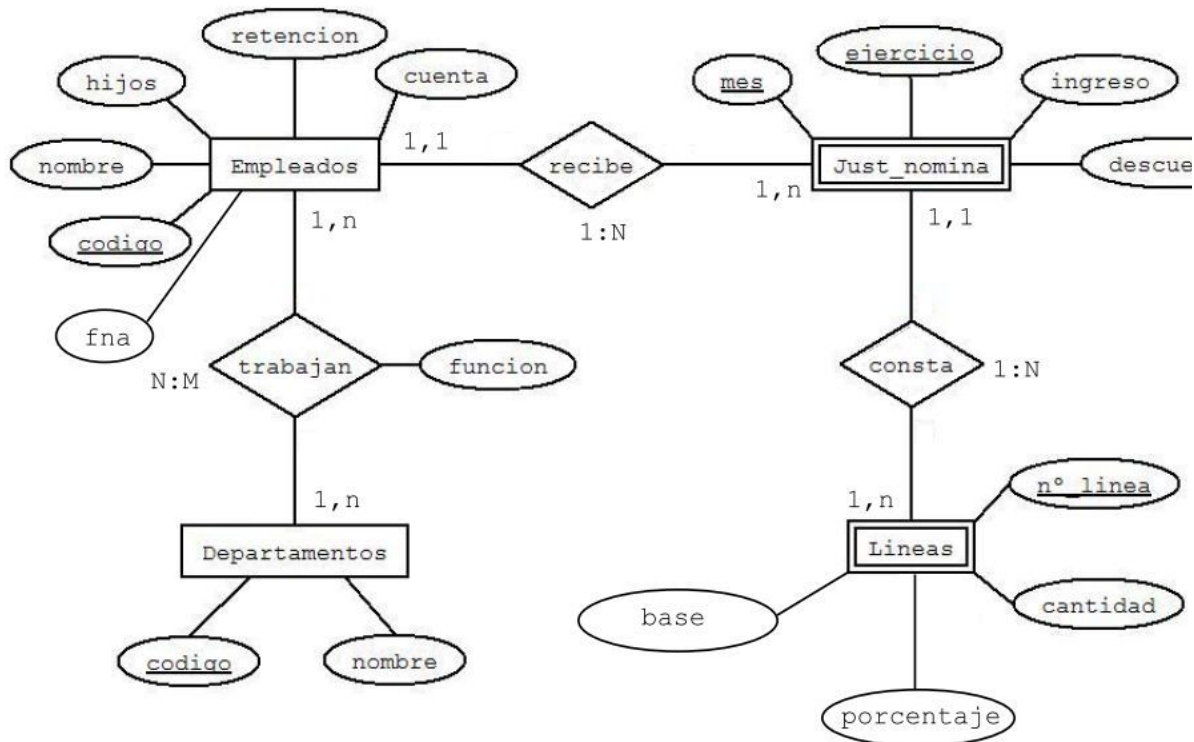
## Enunciado.

La empresa **El Desván**, que se dedica a la rama textil ha decidido informatizar su gestión de nóminas. Para ello **BK programación** desarrollará para ellos la base de datos.

El **gerente** le ha explicado cómo funciona la gestión de nóminas y **Juan**, que será quien se encargue de crear el modelo, las tablas y las consultas, ha recogido la siguiente información:

- A cada empleado se le entrega un justificante de nómina al mes. De cada empleado registramos su código de empleado, nombre, apellidos, número de hijos, cuenta corriente y porcentaje de retención para Hacienda.
- Un empleado puede trabajar en varios Departamentos y en cada uno de ellos realizará una función.
- De un Departamento mantenemos el nombre del mismo y un código de Departamento.
- Los datos de un justificante de nómina son el ingreso total percibido por el empleado y el descuento total aplicado.
- La distinción entre dos justificantes de nómina se hace, además de mediante el código de empleado, mediante el ejercicio fiscal y número de mes al que pertenece.
- Cada justificante de nómina consta de varias líneas y cada línea se identifica por un número de línea del correspondiente justificante. Una línea puede corresponder a un ingreso o a un descuento. En ambos casos se recoge la cantidad (positiva o negativa). En el caso de los descuentos se recoge la base y el porcentaje.

Con todos estos datos ha llegado al siguiente modelo entidad-relación:



También ha creado las bases y ha insertado algunos datos para realizar pruebas de las consultas que haga.

Se procede a crear las bases de datos, realizando la importación desde la consola de comandos:

## Paso 1: Se cargan las tablas:

4.A. Realización de consultas: Ejerc...

Not secure | mecd.es/cidead/aulavirtual/mod/book/view.php?id=37103&chapterid=708

Aula Virtual CIERD(CIDEAD) Mis cursos Recursos Enlaces de interés JUAN M. MORILLO

# Bases

Página Principal

1\_DAM

## TABLA DE CONTENIDO

- 1 La sentencia SQL
- 1.1 Cláusula FROM
- 1.2 Ejercicio 1
- 1.3 Cláusula WHERE
- 1.4 Cláusula ORDER BY
- 1.5 Ordenamiento
- 1.6 Ejercicio 2
- 2 Operadores
- 2.1 Operadores de comparación
- 2.2 Ejercicio 3
- 2.3 Operadores aritméticos y de concatenación
- 2.4 Operadores lógicos
- 2.5 Ejercicio 4
- 2.6 Precedencia
- 3 Consultas calculadas

Ejecutar Línea de Comandos SQL

```
SQL> conn morillobd04
Introduzca la contraseña:
ERROR:
ORA-01017: invalid username/password; logon denied

SQL> conn morillobd04
SP2-0734: inicio "morillobd04" de comando desconocido - resto de la línea ignorado.
SQL> conn morillobd04
Introduzca la contraseña:
Conectado.
SQL> start "C:\Users\ysabe\Documents\CFG5 Desarrollo de Aplicaciones Multiplataforma\Primer Curso\Bases de Datos\UT04 Realización de consultas\Recursos\ScriptsTarea4\tablasTarea4.sql";
Tabla creada.
Tabla creada.
Tabla creada.
Tabla creada.
Tabla creada.
SQL> start
```

Enlace a la solución

Si no estamos seguros de si el contenido del campo está guardado en mayúsculas o minúsculas podemos utilizar las funciones de cadenas de caracteres upper o lower, por ejemplo:

```
select nombre_trab
from trabajos
where upper(NOMBRE_TRAB) not like '%INGENIERO%';
```

Con upper(NOMBRE\_TRAB) obtenemos el nombre en mayúsculas independientemente de como se guardó en la base de datos. Así nos servirá cualquier caso: Ingeniería, ingeniero, INGENIERO, etc.

Igualmente podemos utilizar lower(NOMBRE\_TRAB) y comparar con '%ingeniero%'

**Screenshot Captor Information**

This program is FREE for individual use, and can be used forever, without payment.

However, in order to encourage people to visit DonationCoder.com at least once, we require that you visit our site and download a Freeware or Full Member license key to eliminate this message.

Click the link below to visit our site and generate a freeware license key or make a donation.

For larger companies we offer reasonable commercial site-license rates.

[Visit Website](#)

20:15 26/01/2020

Paso 2: se procede a cargar los datos:

The screenshot shows a web browser window with the URL `meed.es/cidead/aulavirtual/mod/book/view.php?id=37103&chapterid=708`. The page header includes "Aula Virtual CIERD(CIDEAD)", navigation links, and the user name "JUAN M. MORILLO".

Overlaid on the page is a terminal window titled "Ejecutar Línea de Comandos SQL". The terminal output shows the following commands and responses:

```
SQL*Plus: Release 10.2.0.1.0 - Production on Dom Ene 26 20:10:40 2020
Copyright (c) 1982, 2005, Oracle. All rights reserved.

SQL> conn morillobd04
Introduzca la contraseña:
ERROR:
ORA-01017: invalid username/password; login denied

SQL> morillobd04
SP2-0734: inicio "morillobd04" de comando desconocido - resto de la línea ignorado.

SQL> conn morillobd04
Introduzca la contraseña:
Conectado.

SQL> start "C:\Users\ysabe\Documents\CFG5 Desarrollo de Aplicaciones Multiplataforma\Primer Curso\Bases de Datos\UT04 Realización de consultas\Recursos\ScriptsTarea4\tablasTarea4.sql";
Tabla creada.
Tabla creada.
Tabla creada.
Tabla creada.
Tabla creada.
```

The background page displays a table of contents for "Bases de Datos" with sections like "1 La sentencia SQL", "2 Operadores", and "3 Consultas calculadas". A sidebar on the right contains a search bar and a list of items.

A continuación se muestra la pantalla con todas las tablas creadas:

The screenshot displays the Oracle Database Express Edition web interface. The browser address bar shows the URL: 127.0.0.1:8080/apex/f?p=4500:1001:3104802807581468::NO:RP:OB\_CURRENT\_TYPE:TABLE. The page title is 'ORACLE Database Express Edition'. The user is logged in as 'MORILLOBO04'. The main content area shows the 'DEPARTAMENTOS' table structure. The table has two columns: 'CODIGO' (NUMBER(5,0)) and 'NOMBRE' (VARCHAR2(20)). The 'CODIGO' column is the primary key. The table is located in the 'DEPARTAMENTOS' schema. The interface includes a sidebar with a list of tables: DEPARTAMENTOS, EMPLEADOS, JUST\_NOMINAS, LINEAS, and TRABAJAN. The bottom of the screen shows the Windows taskbar with the search bar and various application icons.

Nombre De Columna	Tipo De Dato	Nulo	Valor Por Defecto	Clave Primaria
CODIGO	NUMBER(5,0)	No	-	1
NOMBRE	VARCHAR2(20)	No	-	-

Todo está listo para que le ayudes. Estas son las consultas que debes crear en Oracle 11g Express:

1. Código y nombre de todos los departamentos.

**SELECT CODIGO, NOMBRE FROM DEPARTAMENTOS;**

2. Mes y ejercicio de los justificantes de nómina pertenecientes al empleado cuyo código es 1.

**SELECT MES, EJERCICIO FROM JUST\_NOMINAS WHERE COD\_EMP=1;**

3. Número de cuenta y nombre de los empleados cuya retención es mayor o igual que 10.

**SELECT CUENTA, NOMBRE FROM EMPLEADOS WHERE RETENCION>=10;**

4. Código y nombre de los empleados ordenados ascendentemente por nombre.

**SELECT CODIGO, NOMBRE FROM EMPLEADOS ORDER BY NOMBRE ASC;**

5. Nombre de los empleados que tienen más de 2 hijos.

**SELECT NOMBRE FROM EMPLEADOS WHERE HIJOS>2;**

6. Código y número de cuenta de los empleados cuyo nombre empiece por 'A' o por 'J'.

**SELECT CODIGO, CUENTA FROM EMPLEADOS WHERE NOMBRE LIKE 'A%' OR NOMBRE LIKE 'J%';**

7. Número de empleados que hay en la base de datos.

**SELECT COUNT (\*) "Número total de empleados" FROM EMPLEADOS;**

8. Nombre del primer y último empleado en términos alfabéticos.

**SELECT NOMBRE FROM (SELECT NOMBRE FROM EMPLEADOS ORDER BY NOMBRE ASC) WHERE ROWNUM=1 UNION SELECT NOMBRE FROM (SELECT NOMBRE FROM EMPLEADOS ORDER BY NOMBRE DESC) WHERE ROWNUM=1;**

9. Nombre y número de hijos de los empleados cuya retención es: 8, 10 o 12.

**SELECT NOMBRE, HIJOS FROM EMPLEADOS WHERE RETENCION=8 OR RETENCION=10 OR RETENCION=12;**

10. Número de hijos y número de empleados agrupados por hijos, mostrando sólo los grupos cuyo número de empleados sea mayor que 1.

**SELECT HIJOS, COUNT (\*) as CODIGO FROM EMPLEADOS WHERE CODIGO>1  
GROUP BY HIJOS;**

11. Número de hijos, retención máxima, mínima y media de los empleados agrupados por hijos.

**SELECT HIJOS, MAX (RETENCION), MIN (RETENCION), AVG (RETENCION) FROM  
EMPLEADOS GROUP BY HIJOS;**

12. Nombre y función de los empleados que han trabajado en el departamento 1.

**SELECT EMPLEADOS.NOMBRE, TRABAJAN.FUNCION FROM EMPLEADOS,  
TRABAJAN WHERE TRABAJAN.COD\_DEP=1;**

13. Nombre del empleado, nombre del departamento y función que han realizado de los empleados que tienen 1 hijo.

**SELECT EMPLEADOS.NOMBRE "Nombre Empleados",  
DEPARTAMENTOS.NOMBRE "NOMBRE DEPARTAMENTO",  
TRABAJAN.FUNCION FROM EMPLEADOS INNER JOIN TRABAJAN ON  
(EMPLEADOS.CODIGO=TRABAJAN.COD\_EMP) INNER JOIN DEPARTAMENTOS  
ON (TRABAJAN.COD\_DEP=DEPARTAMENTOS.CODIGO) WHERE  
EMPLEADOS.HIJOS=1;**

14. Nombre del empleado y nombre del departamento en el que han trabajado empleados que no tienen hijos.

**SELECT EMPLEADOS.NOMBRE, DEPARTAMENTOS.NOMBRE  
"DEPARTAMENTO" FROM EMPLEADOS INNER JOIN TRABAJAN ON  
(EMPLEADOS.CODIGO=TRABAJAN.COD\_EMP) INNER JOIN DEPARTAMENTOS  
ON (TRABAJAN.COD\_DEP=DEPARTAMENTOS.CODIGO) WHERE  
EMPLEADOS.HIJOS<=0;**

15. Nombre del empleado, mes y ejercicio de sus justificantes de nómina, número de línea y cantidad de las líneas de los justificantes para el empleado cuyo código=1.

```
SELECT EMPLEADOS.NOMBRE, JUST_NOMINAS.MES,  
JUST_NOMINAS.EJERCICIO, LINEAS.NUMERO, LINEAS.CANTIDAD FROM  
EMPLEADOS INNER JOIN JUST_NOMINAS ON  
(EMPLEADOS.CODIGO=JUST_NOMINAS.COD_EMP) INNER JOIN LINEAS ON  
(EMPLEADOS.CODIGO=LINEAS.COD_EMP) WHERE EMPLEADOS.CODIGO=1;
```

16. Nombre del empleado, mes y ejercicio de sus justificantes de nómina para los empleados que han trabajado en el departamento de Ventas.

```
SELECT EMPLEADOS.NOMBRE, JUST_NOMINAS.MES,  
JUST_NOMINAS.EJERCICIO FROM EMPLEADOS INNER JOIN JUST_NOMINAS  
ON (EMPLEADOS.CODIGO=JUST_NOMINAS.COD_EMP) INNER JOIN TRABAJAN  
ON (EMPLEADOS.CODIGO=TRABAJAN.COD_EMP) INNER JOIN  
DEPARTAMENTOS ON (TRABAJAN.COD_DEP=DEPARTAMENTOS.CODIGO)  
WHERE DEPARTAMENTOS.NOMBRE='Ventas';
```

17. Nombre del empleado e ingresos totales percibidos agrupados por nombre.

```
SELECT EMPLEADOS.NOMBRE, SUM (JUST_NOMINAS.INGRESO) FROM  
EMPLEADOS INNER JOIN JUST_NOMINAS ON  
(EMPLEADOS.CODIGO=JUST_NOMINAS.COD_EMP) GROUP BY  
EMPLEADOS.NOMBRE;
```

18. Nombre de los empleados que han ganado más de 2000 € en el año 2006.

```
SELECT EMPLEADOS.NOMBRE FROM EMPLEADOS INNER JOIN  
JUST_NOMINAS ON (EMPLEADOS.CODIGO=JUST_NOMINAS.COD_EMP) WHERE  
JUST_NOMINAS.EJERCICIO=2006 GROUP BY EMPLEADOS.NOMBRE HAVING  
SUM (JUST_NOMINAS.INGRESO)>2000;
```



19. Número de empleados cuyo número de hijos es superior a la media de hijos de los empleados.

**SELECT COUNT (\*) FROM EMPLEADOS WHERE (SELECT AVG (HIJOS) FROM EMPLEADOS)<HIJOS;**

20. Nombre de los empleados que más hijos tienen o que menos hijos tienen.

**SELECT NOMBRE AS NOMBRE\_EMPLEADOS, HIJOS AS TOTAL\_HIJOS FROM EMPLEADOS WHERE HIJOS=(SELECT MAX (HIJOS) FROM EMPLEADOS) OR HIJOS=(SELECT MIN (HIJOS) FROM EMPLEADOS);**

21. Nombre de los empleados que no tienen justificante de nóminas.

**SELECT NOMBRE FROM EMPLEADOS FULL OUTER JOIN JUST\_NOMINAS ON (EMPLEADOS.CODIGO=JUST\_NOMINAS.COD\_EMP) WHERE INGRESO IS NULL;**

22. Nombre y fecha de nacimiento de todos los empleados.

**SELECT NOMBRE, FNAC AS FECHA\_NACIMIENTO FROM EMPLEADOS;**

23. Nombre y fecha de nacimiento con formato "1 de Enero de 2000" y etiquetada la columna como fecha, de todos los empleados.

**SELECT NOMBRE, CONCAT (TO\_CHAR(FNAC, 'DD "de "'),CONCAT (TO\_CHAR(FNAC, 'Month "de " ' ),TO\_CHAR(FNAC, 'YYYY')) AS fecha FROM EMPLEADOS;**

24. Nombre de los empleados, nombre de los departamentos en los que ha trabajado y función en mayúsculas que ha realizado en cada departamento.

**SELECT EMPLEADOS.NOMBRE AS NOMBRE\_EMPLEADO, DEPARTAMENTOS.NOMBRE AS NOMBRE\_DEPARTAMENTO, UPPER (TRABAJAN.FUNCION) FROM EMPLEADOS INNER JOIN TRABAJAN ON**

**(EMPLEADOS.CODIGO=TRABAJAN.COD\_EMP) INNER JOIN DEPARTAMENTOS  
ON (TRABAJAN.COD\_DEP=DEPARTAMENTOS.CODIGO);**

25. Nombre, fecha de nacimiento y nombre del día de la semana de su fecha de nacimiento de todos los empleados.

**SELECT NOMBRE, CONCAT (FNAC, TO\_CHAR(FNAC,' DAY')) AS FECHA FROM  
EMPLEADOS;**

26. Nombre y edad de los empleados.

**SELECT NOMBRE, TRUNC(MONTHS\_BETWEEN (SYSDATE, FNAC)/12) AS  
MI\_EDAD FROM EMPLEADOS;**

27. Nombre, edad y número de hijos de los empleados que tienen menos de 40 años y tienen hijos.

**SELECT NOMBRE, TRUNC(MONTHS\_BETWEEN (SYSDATE, FNAC)/12) AS EDAD,  
HIJOS FROM EMPLEADOS WHERE HIJOS>0 AND TRUNC(MONTHS\_BETWEEN  
(SYSDATE, FNAC)/12)<40;**

28. Nombre, edad de los empleados y nombre del departamento de los empleados que han trabajado en más de un departamento.

**SELECT EMPLEADOS.NOMBRE, TRUNC(MONTHS\_BETWEEN (SYSDATE,  
EMPLEADOS.FNAC)/12) AS EDAD, DEPARTAMENTOS.NOMBRE AS  
DEPARTAMENTOS FROM EMPLEADOS INNER JOIN TRABAJAN ON  
(EMPLEADOS.CODIGO=TRABAJAN.COD\_EMP) INNER JOIN DEPARTAMENTOS  
ON (TRABAJAN.COD\_DEP=DEPARTAMENTOS.CODIGO) WHERE  
EMPLEADOS.CODIGO IN (SELECT COD\_EMP FROM TRABAJAN GROUP BY  
COD\_EMP HAVING COUNT (COD\_DEP)>1);**

29. Nombre, edad y número de cuenta de aquellos empleados cuya edad es múltiplo de 3.

```
SELECT EMPLEADOS.NOMBRE, TRUNC(MONTHS_BETWEEN (SYSDATE,
EMPLEADOS.FNAC)/12) AS EDAD, EMPLEADOS.CUENTA FROM EMPLEADOS
WHERE MOD( TRUNC(MONTHS_BETWEEN (SYSDATE,
EMPLEADOS.FNAC)/12),3)=0;
```

30. Nombre e ingresos percibidos empleado más joven y del más longevo.

```
SELECT EMPLEADOS.NOMBRE, SUM (JUST_NOMINAS.INGRESO) AS
Ingresos_Percibidos FROM EMPLEADOS INNER JOIN JUST_NOMINAS ON
(EMPLEADOS.CODIGO=JUST_NOMINAS.COD_EMP) WHERE EMPLEADOS.FNAC
<= ALL (SELECT FNAC FROM EMPLEADOS) GROUP BY EMPLEADOS.NOMBRE
UNION SELECT EMPLEADOS.NOMBRE, SUM(JUST_NOMINAS.INGRESO) AS
INGRESOS FROM EMPLEADOS JOIN JUST_NOMINAS ON
JUST_NOMINAS.COD_EMP=EMPLEADOS.CODIGO WHERE EMPLEADOS.FNAC
>= ALL (SELECT FNAC FROM EMPLEADOS) GROUP BY EMPLEADOS.NOMBRE;
```

The screenshot shows the Oracle Database Express Edition interface. The SQL Command window contains the following query:

```
SELECT EMPLEADOS.NOMBRE, SUM (JUST_NOMINAS.INGRESO) AS Ingresos_Percibidos FROM EMPLEADOS INNER JOIN JUST_NOMINAS ON (EMPLEADOS.CODIGO=JUST_NOMINAS.COD_EMP) WHERE EMPLEADOS.FNAC <= ALL (SELECT FNAC FROM EMPLEADOS) GROUP BY EMPLEADOS.NOMBRE UNION SELECT EMPLEADOS.NOMBRE, SUM(JUST_NOMINAS.INGRESO) AS INGRESOS FROM EMPLEADOS JOIN JUST_NOMINAS ON JUST_NOMINAS.COD_EMP=EMPLEADOS.CODIGO WHERE EMPLEADOS.FNAC >= ALL (SELECT FNAC FROM EMPLEADOS) GROUP BY EMPLEADOS.NOMBRE;
```

The results are displayed in a table with two columns: NOMBRE and INGRESOS\_PERCIBIDOS. The table contains two rows of data:

NOMBRE	INGRESOS_PERCIBIDOS
Fernando Romero D'Áas	5000
Rosa Lorte Lopez	4800

Below the table, it indicates "2 filas devueltas en 0,01 segundos" and provides a link for "Exportación de CSV". The bottom status bar shows the application version as "Application Express 2.1.0.00.39" and the copyright as "Copyright © 1999, 2008, Oracle. Todos los Derechos Reservados".

Nota, no he puesto imagen de todas las consultas, porque sería un trabajo muy largo, todas las consultas han sido probada.

