

FEB. 2018

BASES de DATOS

TAREFA_04

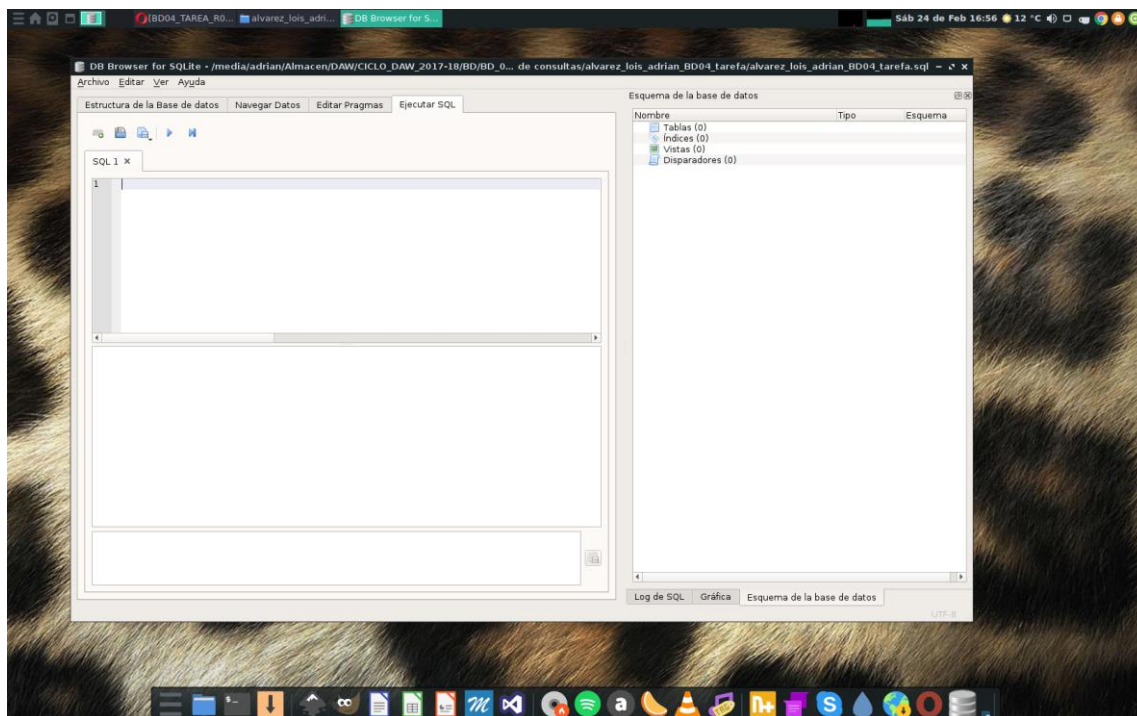
REALIZACIÓN DE CONSULTAS_UD04

DAW 2017/18
ADRIÁN ÁLVAREZ LOIS



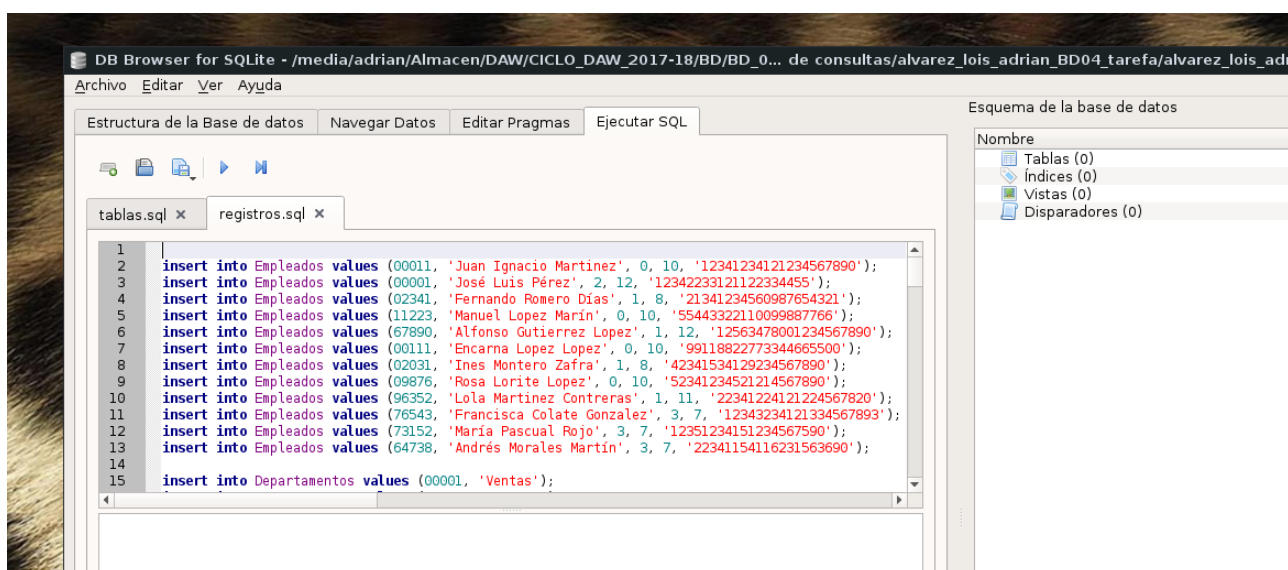
BD - TAREFA UNIDADE DIDÁCTICA 4

Primeiramente empreguei o programa DB Browser for SQLite en Debian Linux porque Oracle e Linux Debian non se levan moi ben. A Librería SQLite permite traballar con bases de datos sen ter que conectarse a un servidor para bases de datos pequenas con non moitas instrucións de código e corre moi ben no meu Debian 9.3.

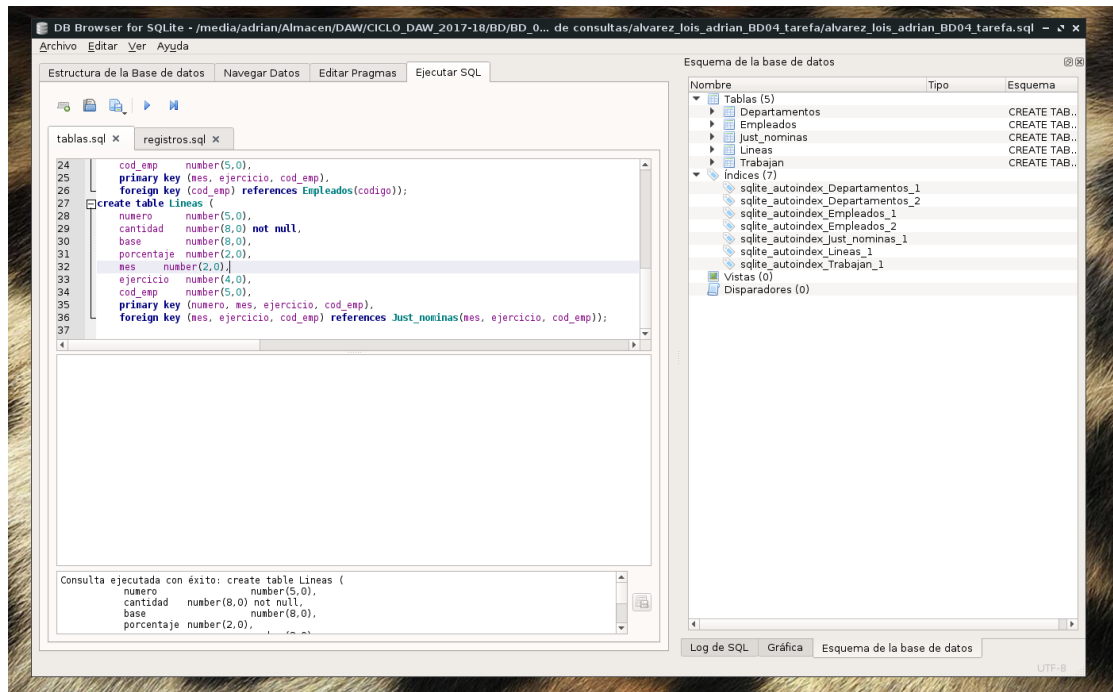


Despois de iniciar o programa vemos á dereita o esquema de táboas e á esquerda un espazo para introducir código SQL arriba e visualizar a execución debaixo.

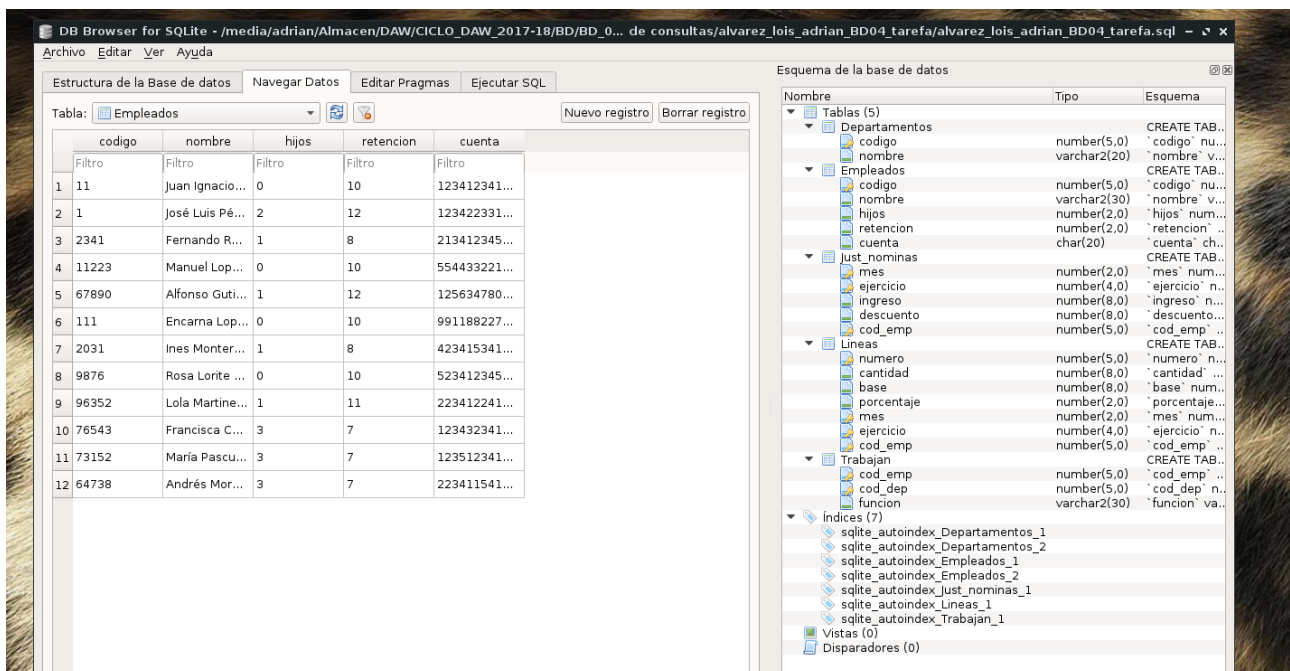
Creamos unha base de datos chamada tarefa04.sql e importamos os scripts tablas.sql e registros.sql.



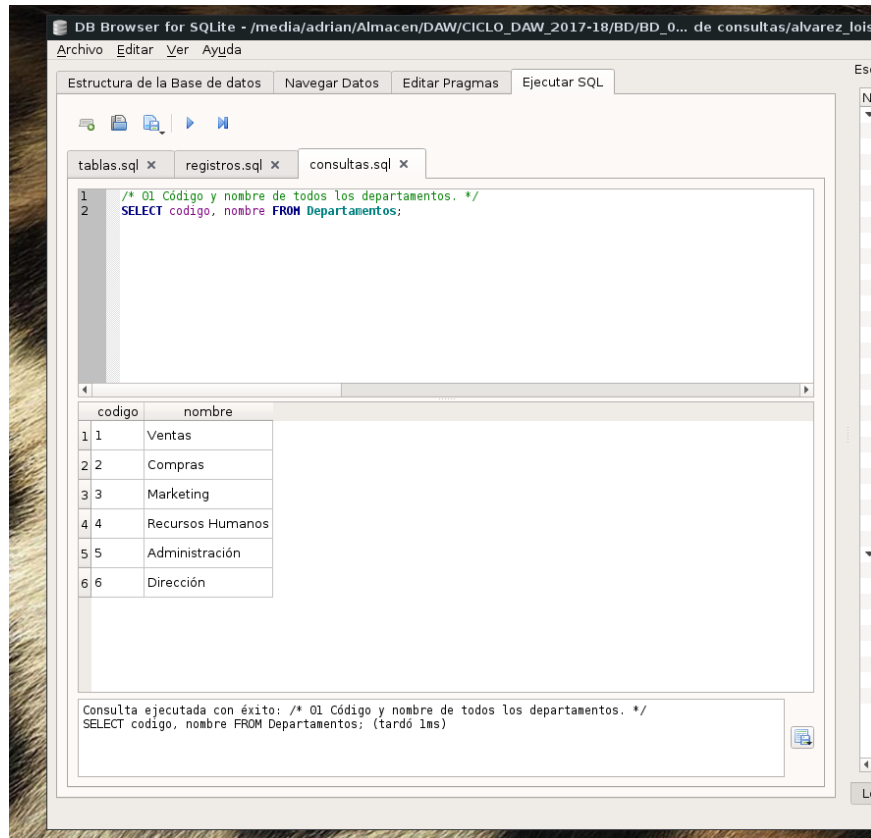
Basta con executar os scripts para que se carguen as táboas e os rexistros. Cabe sinalar que o limitado da biblioteca SQLite fai que xurdan problemas de sintaxe e coas funcións de manipulación de datas polo que despois me verei obrigado a mudarme a Oracle SQL Developer en Windows 10.



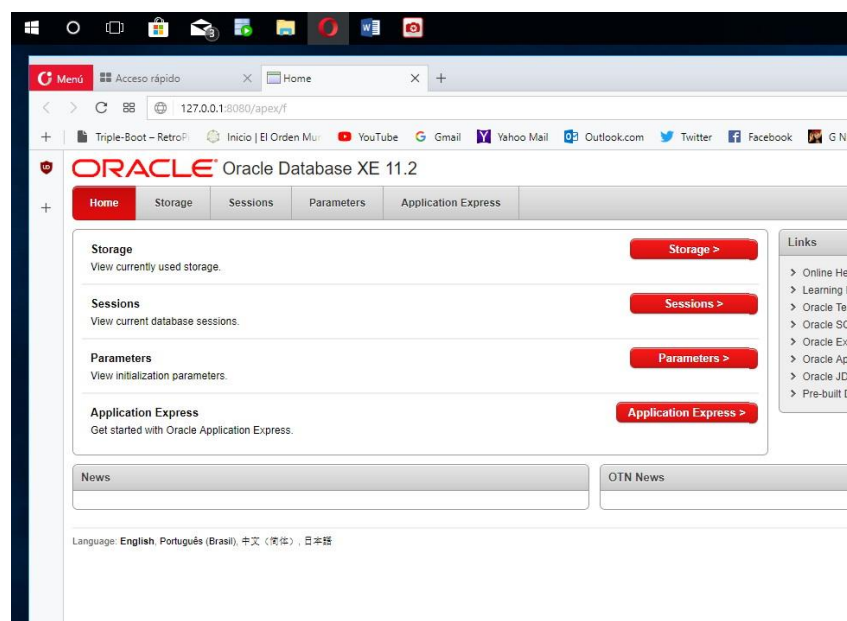
Algo que me gusta moito do DB Browser for SQLite é que podes navegar e visualizar os rexistros das táboas. Non cargou o campo FNacimiento na táboa Empleados por limitacións de SQLite.



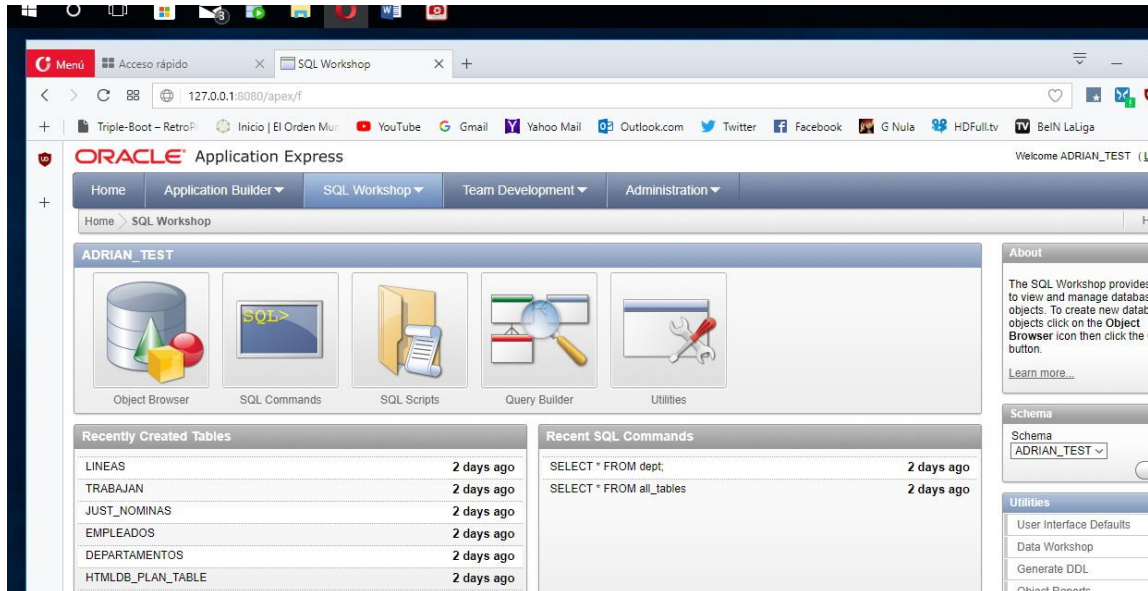
Coas táboas e os rexistros cargados na miña base de datos abro unha lapela nova para introducir código SQL e creo un arquivo chamado consultas.sql onde vou ir escribindo o código das consultas do enunciado e executando as liñas para ver os resultados.



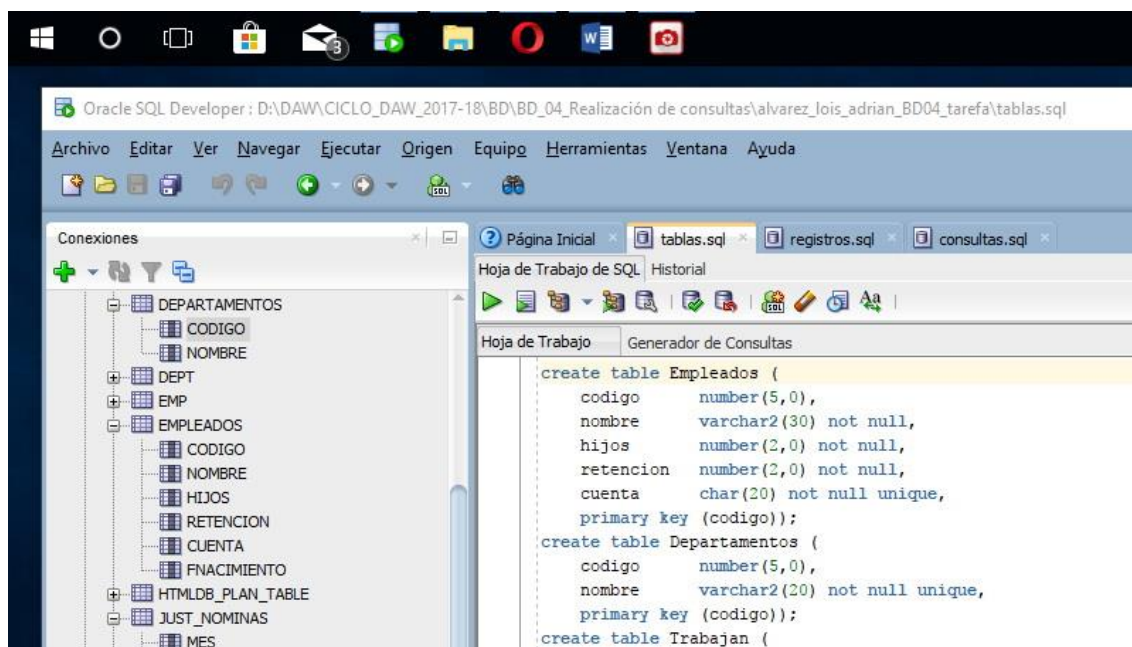
Estiven realizando as primeiras consultas así e moi ben pero cando non me recoñeceu OUTER JOIN e me decatei dos problemas coas datas mudei a Windows 10 usando Oracle SQL Developer.



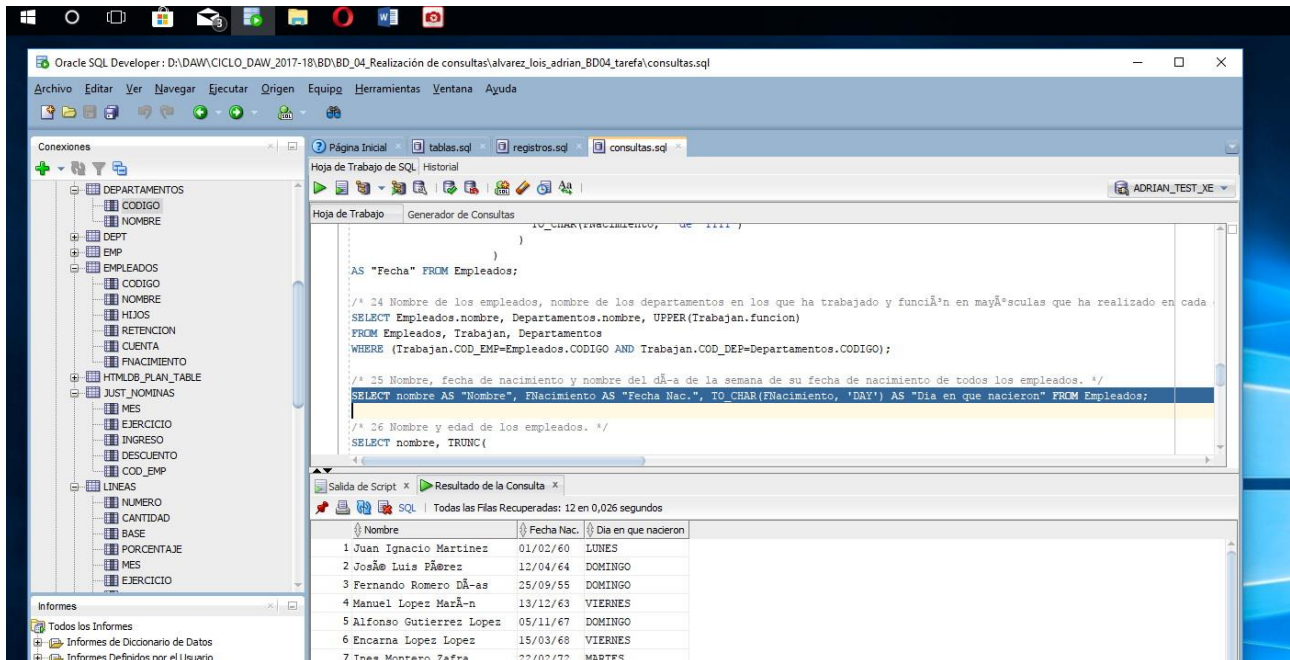
Coas Express Tools de Oracle que xa tiña instaladas, accedín a Application Express e usei un Workspace que xa tiña chamado ADRIAN_TEST co que me conectarei ao servidor de oracle.



Eu persoalmente usei SQL Developer para conectarme ao servidor de Oracle. Logo abrín os scripts e ao executalos xa se cargaron as táboas e os rexistros no servidor.



Unha vez con todo listo é hora de abrir os script consultas.sql e completar a tarefa e como en Linux podó comprobar a execución do código.



A continuación amoso o contido do ficheiro consultas.sql onde se resposta ao enunciado da tarefa.

```
-- DAW 2017/18
-- BASES DE DATOS
-- TAREFA UD 04
-- Autor: Adrian Alvarez Lois
```

```
/* 01 Codigo y nombre de todos los departamentos. */
```

```
SELECT codigo, nombre FROM Departamentos;
```

```
/* 02 Mes y ejercicio de los justificantes de nomina pertenecientes al  
empleado cuyo codigo es 1. */
```

```
SELECT mes, ejercicio FROM just_nominas where cod_emp = 1;
```

```
/* 03 Numero de cuenta y nombre de los empleados cuya retencion es mayor  
o igual que 10. */
```

```
SELECT cuenta, nombre FROM Empleados WHERE retencion >= 10;
```

```
/* 04 Codigo y nombre de los empleados ordenados ascendentemente por nombre.  
*/
```

```
SELECT codigo, nombre FROM Empleados ORDER BY nombre ASC;
```

/* 05 Nombre de los empleados que tienen mas de 2 hijos.*/

SELECT nombre FROM Empleados where hijos > 2;

/* 06 Codigo y numero 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%';

/* 07 Numero de empleados que hay en la base de datos. */

SELECT COUNT (nombre) FROM Empleados;

/* 08 Nombre del primer y Ultimo empleado en terminos alfabeticos. */

SELECT MIN (nombre), MAX (nombre) FROM Empleados ORDER BY nombre;

/* 09 Nombre y numero de hijos de los empleados cuya retencion es: 8, 10 o 12. */

SELECT nombre, hijos FROM Empleados WHERE retencion IN (8,10,12);

/* 10 Numero de hijos y numero de empleados agrupados por hijos, mostrando solo los grupos cuyo numero de hijos sea mayor que 1.*/

SELECT hijos, COUNT (*) FROM Empleados WHERE hijos > 1 GROUP BY hijos;

/* 11 Numero de hijos, retencion maxima, minima y media de los empleados agrupados por hijos. */

SELECT hijos, MAX (retencion), MIN (retencion), AVG (retencion) FROM Empleados GROUP BY hijos;

/* 12 Nombre y funcion de los empleados que han trabajado en el departamento 1. */

SELECT Empleados.nombre, Trabajan.funcion FROM Empleados JOIN Trabajan ON Trabajan.cod_emp = Empleados.codigo WHERE Trabajan.cod_dep = 1;

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

SELECT Empleados.nombre, Departamentos.nombre, Trabajan.funcion FROM Departamentos, Empleados, TrSELECT Empleados.nombre, Just_nominas.mes, Just_nominas.ejercicio FROM Just_nominas, Empleados JOIN Departamentos ON Departamentos.codigo=Trabajan.cod_dep WHERE Departamentos="Ventas"; abajan WHERE (Trabajan.cod_dep=Departamentos.codigo AND Trabajan.cod_emp=Empleados.codigo) AND 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 FROM Empleados, Trabajan,
Departamentos WHERE (Trabajan.cod_dep=Departamentos.codigo AND
Trabajan.cod_emp=Empleados.codigo) AND Empleados.hijos=0;
```

/* 15 Nombre del empleado, mes y ejercicio de sus justificantes de nomina, numero de linea y cantidad de las líneas de los justificantes para el empleado cuyo codigo=1.*/

```
SELECT Empleados.nombre, Lineas.mes, Lineas.ejercicio, Lineas.numero,
Lineas.cantidad
FROM Empleados, Lineas WHERE (Empleados.codigo=Lineas.cod_emp AND
Empleados.codigo = 1) ORDER BY Lineas.mes ASC, Lineas.ejercicio DESC;
```

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

```
SELECT Empleados.nombre, Just_nominas.mes, Just_nominas.ejercicio FROM
Empleados, Just_nominas, Trabajan, Departamentos WHERE
(Departamentos.codigo=Trabajan.cod_dep AND
Empleados.codigo=Trabajan.cod_emp AND Departamentos.nombre='Ventas')
ORDER BY Just_nominas.ejercicio DESC, Just_nominas.mes DESC ;
```

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

```
SELECT empleados.nombre, SUM(Just_nominas.ingreso) FROM Empleados JOIN
Just_nominas ON Just_nominas.cod_emp=Empleados.codigo GROUP BY
Empleados.nombre;
```

/* 18 Nombre de los empleados que han ganado mas de 2000 \$ en el año 2006. */

```
SELECT Empleados.nombre FROM Empleados JOIN Just_nominas ON
(Just_nominas.cod_emp=Empleados.codigo) WHERE Just_nominas.ejercicio=2006
GROUP BY Empleados.nombre HAVING SUM(Just_nominas.ingreso)>2000 ;
```

/* 19 Numero de empleados cuyo numero de hijos es superior a la media de hijos de los empleados. SUBCONSULTAS */

```
SELECT COUNT(codigo) FROM Empleados WHERE hijos >(SELECT AVG(hijos) FROM
Empleados);
```

/* 20 Nombre de los empleados que mas hijos tienen o que menos hijos tienen. CONSULTAS MULTITABLA Y SUBCONSULTAS */

```
SELECT nombre FROM Empleados WHERE hijos = (SELECT MAX(hijos) FROM
Empleados)
UNION
SELECT nombre FROM Empleados WHERE hijos = (SELECT MIN(hijos) FROM
Empleados);
```


/* 21 Nombre de los empleados que no tienen justificante de nominas. */

```
SELECT Empleados.nombre FROM Empleados FULL OUTER JOIN Just_nominas ON
(Just_nominas.cod_emp=Empleados.codigo) WHERE Just_nominas.ingreso IS NULL;
```

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

```
SELECT nombre, fnacimiento 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, TO_CHAR(fnacimiento,'DD "de" MONTH "de" YYYY') AS fecha
FROM Empleados;
```

```
-- FUNCIONES DE CADENA DE CARACTERES y DE CONVERSIÓN
-- Consulta Buena. Me costó pillarlo.
-- Tuve que tabular el codigo para no hacerme un lío
```

```
SELECT Nombre, CONCAT(
    TO_CHAR(FNacimiento, 'DD "de "')
    ,
    CONCAT(
        INITCAP(TO_CHAR(FNacimiento,'MONTH'))
        ,
        TO_CHAR(FNacimiento,' "de "YYYY')
    )
)
AS "Fecha" FROM Empleados;
```

/* 24 Nombre de los empleados, nombre de los departamentos en los que ha trabajado y funcion en mayusculas que ha realizado en cada departamento.*/

```
SELECT Empleados.nombre, Departamentos.nombre, UPPER(Trabajan.funcion)
FROM Empleados, Trabajan, Departamentos
WHERE (Trabajan.COD_EMP=Empleados.CODIGO AND
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 AS "Nombre", FNacimiento AS "Fecha Nac.", TO_CHAR(FNacimiento,
'DAY') AS "Dia en que nacieron" FROM Empleados;
```

/* 26 Nombre y edad de los empleados. */

```
SELECT nombre, TRUNC(
    MONTHS_BETWEEN(SYSDATE,FNacimiento)/12,
    0 -- Podria no poner este factor
)
AS "EDAD"
FROM Empleados;
```

/* 27 Nombre, edad y numero de hijos de los empleados que tienen menos de 40 años y tienen hijos. */

```
SELECT nombre, TRUNC(MONTHS_BETWEEN(SYSDATE, FNacimiento)/12) AS "EDAD",  
hijos  
FROM Empleados  
WHERE hijos > 0 AND MONTHS_BETWEEN(SYSDATE, FNacimiento) < 480  
;
```

/* 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.FNacimiento)/12) AS "EDAD",  
Departamentos.nombre AS "DEPARTAMENTO"  
FROM Empleados, Trabajan, Departamentos  
WHERE (Trabajan.COD_EMP=Empleados.CODIGO AND  
Trabajan.COD_DEP=Departamentos.CODIGO)  
AND Empleados.CODIGO IN (SELECT Cod_Emp FROM Trabajan GROUP BY Cod_Emp  
HAVING COUNT(Cod_Dep)>1) -- Importante crear GROUP BY antes de pasar el  
HAVING44  
;
```

/* 29 Nombre, edad y numero de cuenta de aquellos empleados cuya edad es multiplo de 3.*/

```
SELECT nombre, TRUNC(MONTHS_BETWEEN(SYSDATE, Empleados.FNacimiento)/12) AS  
"EDAD", cuenta  
FROM Empleados  
WHERE MOD(TRUNC(MONTHS_BETWEEN(SYSDATE, FNacimiento)/12), 3)=0 ;  
-- WHERE (TRUNC(MONTHS_BETWEEN(SYSDATE, FNacimiento)/12)/3)=1;??????????
```

/* 30 Nombre e ingresos percibidos empleado mas joven y del mas longevo. */

```
SELECT Empleados.nombre, SUM(Just_nominas.ingreso) AS "INGRESOS"  
FROM Empleados JOIN Just_nominas ON (Just_nominas.Cod_Emp=Empleados.Codigo)  
WHERE Empleados.FNacimiento <= ALL (SELECT FNacimiento FROM Empleados) --  
SUBCONSULTAS cuyo resultado es más de una fila.  
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.FNacimiento >= ALL (SELECT FNacimiento FROM Empleados)  
WHERE Empleados.FNacimiento = (SELECT MAX(FNacimiento) FROM Empleados) --  
Otra forma de hacerlo  
GROUP BY Empleados.Nombre  
;
```