select \* from doctor;

select \* from emp where dept\_no=10;

-- mostrar todos los empleados cuyo oficios sea director

-- todo lo que no sea una nº se escribe entre comillas simples

-- Oracle diferencia entre mayúsculas y minúsculas

select \* from EMP where OFICIO='DIRECTOR';

select \* from EMP where OFICIO='director';

/\* OPERADORES RELACIONALES

PERMITEN PREGUNTAR POR MÁS DE UN FILTRO EN UNA MISMA CONSULTA

and TODAS LAS CONDICIONES DEBEN CUMPLIRSE

or MUESTRA DATOS DE CADA CONDICIÓN

not PROHIBIDO NO LO DEBEMOS UTILIZAR. NIEGA UNA CONDICIÓN

\*/

-- Mostrar todos los empleados del departamento 10 y el departamento 20

select \* from EMP where DEPT\_NO=10 or DEPT\_NO=20;

-- Mostrar todos los empleados cuyo EMP\_NO esté entre 7800 y 7900

select \* from EMP where EMP\_NO>= 7800 and EMP\_NO<= 7900;

-- Mostrar todos los empleados que NO sean vendedores

select \* from EMP where not OFICIO='VENDEDOR'; -- PROHIBIDO porque va en positivo a los que sí son vendedor y luego los cambia

select \* from EMP where OFICIO <> 'VENDEDOR'; -- éste es el más óptimo, va directamente a por lo que solicitamos

-- Existen otros tipos de operadores para filtrar

-- Operador Between: busca entre un rango y devuelve los valores inclusive del rango

-- Mostrar los empleados cuyo salario esté entre 318000 y 390000

select \* from EMP where SALARIO between 318000 and 390000;

--también así, es lo mismo

select \* from EMP where SALARIO >= 318000 and SALARIO <= 390000;

-- OPERADOR IN

-- Busca en un mismo campo/columna múltiples valores de igualdad

-- Mostrar todos los empleados del departamento 10, 20, 30... así lo puedo pedir sucesivamente

-- CAMPO IN (VALOR 1, VALOR2, VALOR3)

select \* from EMP where DEPT\_NO = 10 or DEPT\_NO = 20 or DEPT\_NO = 30;

select \* from EMP where DEPT\_NO in (10,20,30,55,66,88); -- es igual a la anterior y la lista la puedes hacer más grande, muy útil

-- OPERADOR NOT IN

-- Muestra los contrarios al operador in (todos los que no estén en un rango)

-- Mostrar todos los empleados que no estén en el departamento 10 ni en el 20

select \* from EMP where DEPT\_NO not in (10, 20); -- en este caso es más eficiente

-- OPERADOR LIKE

-- SE UTILIZA PARA BUSCAR COINCIDENCIAS EN TEXTOS (STRING, VARCHAR)

-- BUSCADORES

-- UTILIZA UNA SERIE DE CARACTERES ESPECIALEWS (SOLO SIRVE UNO)

-- ? Busca un dígito en el texto

-- \_ Representa un caracter cualquiera

-- % Busca cualquier caracter y cualquier longitud

-- Mostrar todos los empleados cuyo apellido comience con la S

select \* from EMP where apellido like 's%';

-- Mostrar todos los empleados cuyo apellido comience con la S y termine con a

select \* from EMP where apellido like 's%a';

-- Mostrar todos los empleados sea de 4 letras

select \* from EMP where apellido like '\_\_\_\_'; -- sirve para buscar en los textos

-- CLAUSULA DISTINCT

-- Nos permite quitar repetidos de una consulta Select

-- Se utiliza cuando hay pocos datos

-- Mostrar los oficios de los empleados

select OFICIO from EMP; -- así los vemos de tooooodos los empleados

-- quito los diferentes

select distinct OFICIO from EMP;

-- campos calculados

-- Un campo calculado es una columna que NO existe en una tabla

-- pero se genera a partir de otra columna

-- OPERADORES MATE MÁTICOS +,-,/,\*

-- mostrar el apellido y el salario total (salario + comisión) de todos los empleados-

select APELLIDO, (SALARIO+COMISION) as Total from EMP; -- La consulta la guardo en un alias con "AS"

where (SALARIO + COMISION) >= 210000;

-- Todo campo calculado

-- Los campos calculados no se pueden filtrar con where

--Un where solamente se uitliza para filtrar datos de la tabla

--Un order by se ejecuta sobre el cursos

select APELLIDO, (SALARIO + COMISION) as TOTAL from EMP order by TOTAL asc;

-- 1. Mostrar todos los datos de los empleados de nuestra tabla EMP

select \* from EMP;

--2.- Mostrar el apellido, oficio, salario anual, con las dos extras para aquellos empleados con comisión mayor de 100000

select APELLIDO, OFICIO, (SALARIO\*14 + COMISION) as SALARIO\_ANUAL from EMP where COMISION >100000;

-- 3.- Idem del anterior pero para aquellos empleados que sumen entre salario anual con extras y comision el 750000

select APELLIDO, OFICIO, (SALARIO\*14 + COMISION)as SALARIO\_ANUAL from EMP where SALARIO\_ANUAL > 750000;

--4.- Idem del anterior pero para aquellos empleados que sumen entre salario anual con extras y comision el 1.000.000.

select APELLIDO, OFICIO, (SALARIO\*14 + COMISION) as SALARIO\_ANUAL from EMP where SALARIO\_ANUAL = 1000000;

--5.- Mostrar todos los datos de los empleados ordenador por departamento y dentro de este por oficio para tener una visión

-- jerárquica . (order by)

select \* from EMP order by DEPT\_NO, OFICIO;

-- 6.- Mostrar todos los enfermos nacidos antes del 1/1/70

select \* from ENFERMO;

select \* from ENFERMO where FECHA\_NAC < '1/1/70';

-- 7.- Igual que el anterior, para los nacidos antees del 1/1/70 ordenados por número de inscripción

select \* from ENFERMO where FECHA\_NAC < '1/1/70' order by INSCRIPCION;

-- 8.- Listas todos los datos de la plantilla del hospital del turno de mañana

select \* from PLANTILLA;

select \* from PLANTILLA where TURNO = 'M';

-- 9.- IDEM para el turno de noche

select \* from PLANTILLA where TURNO = 'N';

-- 10.- Listar los doctores que su salario anual supere 3000000 ptas.

select \* from DOCTOR;

select \* from DOCTOR where SALARIO >3000000;

-- 11.- Visualizar los empleados de la plantilla del turno de mañana que tengan un salario

-- entre 2000000 y 2250000

select \* from PLANTILLA where TURNO= 'M' and

--12.- Mostrar los nombres de los departamentos situados en Madrid o en Barcelona

—————————————————————————

-- Contar el número de registros de la tabla EMP

select count(\*) as REGISTROS from EMP; -- OJO SIEMPRE DEBE DE LLEVAR ALIAS, YA QUE ES UN CAMPO CALCULADO

-- IDEM Y EL MÁXIMO SALARIO DE LA TABLA

select count(\*) as REGISTROS, MAX(SALARIO) as MAXIMO from EMP;

--PODEMOS MOSTRAR LOS DATOS POR COLUMNAS

--la agrupación de columnas se realiza en el SELECT y debemos indicar mediante

-- group by, cada columna por la que estamos agrupando

-- TRUCO: debemos incluir en el group by cada columna que sea una función

-- de agrupación en el SELECT.

-- Mostrar el número de personas por cada oficio

select count(\*) as PERSONAS, OFICIO from EMP

group by OFICIO;

-- Podemos ordenar por más de un campo

select count(\*) as PERSONAS, OFICIO, DEPT\_NO from EMP

group by OFICIO, DEPT\_NO;

-- Podemos filtrar estas consultas

-- para filtrar tenemos dos opciones:

-- 1.- where (filtra por los datos de la tabla) ANTES DEL GROUP BY

-- 2.- having (filtra por los datos del grupo) DESPUÉS DEL GROUP BY

-- MOSTRAR número de personas director y analistas

select count(\*) as PERSONAS, OFICIO from EMP

where OFICIO in ('DIRECTOR', 'ANALISTA')

group by OFICIO;

-- Podemos hacer lo mismo con Having (OJO AL ORDEN, ES AL REVÉS)

select count(\*) as PERSONAS, OFICIO from EMP

group by OFICIO

having OFICIO in ('DIRECTOR', 'ANALISTA');

-- Por norma, si estoy filtrando por allgún campo de la columna select con GROUP BY

-- es más rápido usar "HAVING"

-- MOSTRAR número de personas POR CADA OFICIO

-- PERO SOLAMENTE DEL DEPARTAMENTO (10 Y 20)

-- Aquí no hay opción, solo se puede hacer así (aquí no puedo usar Having porque dept\_no no está en el select

select count(\*) as PERSONAS, OFICIO from EMP

where DEPT\_NO in (10,20)

group by OFICIO;

-- Mostrar el número de personas por cada oficio pero solamente donde tengamos más de una persona trabajando

select count(\*) as PERSONAS, OFICIO from EMP

group by oficio

having COUNT(\*) >10; -- no se puede usar el alias personas con el having

----- EJERCICIOS DE CONSULTAS DE AGRUPACIÓN --

--1 .-Encontrar el salario medio de los analistas, mostrando el número de los empleados con oficio analista.

-- mío

select avg(salario) as SALARIOMEDIO, count(\*) as PERSONAS, OFICIO from EMP

group by OFICIO having OFICIO = 'ANALISTA';

-- 2.- Encontrar el salario más alto, mas bajo y la diferencia entre ambos de todos los empleados con oficio EMPLEADO.

select oficio, max(salario) as SALARIOMAXIMO

, min(salario) as SALARIOMINIMO

, max(salario) - min(salario) as DIFERENCIA

from emp group by oficio having oficio = 'EMPLEADO';

-- 3.- Visualizar los salarios mayores para cada oficio.

select oficio, max(salario) as SALARIOMAXIMO from emp group by oficio;

-- 4.- Visualizar el número de personas que realizan cada oficio en cada departamento.

select dept\_no,

count(\*) as PERSONAS, oficio

from emp group by dept\_no, oficio

order by 1;

-- 5.- Buscar aquellos departamentos con cuatro o más personas trabajando.

select dept\_no

, count(\*) as PERSONAS from emp

group by dept\_no having count(\*) > 3;

-- 6.- Mostrar el número de directores que existen por departamento.

select count(\*) as NUMEROEMPLEADOS, dept\_no

from emp

where oficio = 'DIRECTOR'

group by dept\_no;

-- 7.- Visualizar el número de enfermeros, enfermeras e interinos que hay en la plantilla, ordenados por la función.

select funcion, count(funcion) as numeroenfermeros from plantilla

group by funcion;

--7bis Visualizar departamentos, oficios y número de personas, para aquellos departamentos que tengan dos o más personas trabajando en el mismo oficio.

select dept\_no

,count(\*) as PERSONAS,

oficio from emp group by dept\_no,oficio

having count(\*) > 1;

-- 8.- Calcular el salario medio, Diferencia, Máximo y Mínimo de cada oficio. Indicando el oficio y el número de empleados de cada oficio.

select oficio, count(\*) as EMPLEADOS

, min(salario) as SALARIOMAXIMO

, max(salario) as SALARIOMINIMO

, max(salario) - min(salario) as DIFERENCIA

, avg(salario) as MEDIA

from emp group by oficio;

-- 9.- Calcular el valor medio de las camas que existen para cada nombre de sala. Indicar el nombre de cada sala y el número de cada una de ellas.

select trunc(avg(num\_cama),2) as VALORMEDIOCAMAS, NOMBRE, SALA\_COD from SALA

group by NOMBRE, SALA\_COD;

-- 10.- Calcular el salario medio de la plantilla de la sala 6, según la función que realizan. Indicar la función y el número de empleados.

select trunc(avg(salario),2) as MEDIA\_SALARIAL, COUNT(\*) AS PERSONAS, FUNCION from plantilla

where sala\_cod = 6

group by FUNCION;

-- 11.- Averiguar los últimos empleados que se dieron de alta en la empresa en cada uno de los oficios, ordenados por la fecha.

select max(fecha\_alt) AS FECHAMAXIMA,

Oficio from emp

group by oficio

order by 1;

-- 12.- Mostrar el número de hombres y el número de mujeres que hay entre los enfermos.

select count(sexo) as PERSONAS, sexo from enfermo

group by sexo;

-- 13.-Mostrar la suma total del salario que cobran los empleados de la plantilla para cada función y turno.

select turno, funcion, sum(salario) as salario\_total from plantilla

group by turno, funcion;

-- 14.-Calcular el número de salas que existen en cada hospital.

select count(\*) as NUMEROSALAS

, hospital\_cod from sala

group by hospital\_cod;

-- 15.- Mostrar el número de enfermeras que existan por cada sala.

select count(\*) as PERSONAS

, sala\_cod, funcion from plantilla

where funcion='ENFERMERA'

group by sala\_cod, Funcion

order by 1;

--- consultas de combinación

-- Es un from de unión

/\* select TABLA1.CAMPO1, TABLA1.CAMPO2,

TABLA2.CAMPO1, TABLA2.CAMPO2

from TABLA1 inner join TABLA2

on TABLA1.CAMPORELACION = TABLA2.CAMPORELACION;

no importa el orden de TABLA1 o TABLA2 en ningún sitio

\*/

-- Mostrar el Apellido, Oficio de los empleados

-- junto a su nombre de departamento y localidad

select emp.apellido, emp.oficio, dept.Dnombre, dept.loc

from EMP

inner join DEPT

on EMP.DEPT\_NO = DEPT.DEPT\_NO;

-- Mostrar el Apellido, Oficio, nombre de departamento y localidad de los empleados

-- de Sevilla

select emp.apellido, emp.oficio, dept.Dnombre, dept.loc

from EMP

inner join DEPT

on EMP.DEPT\_NO = DEPT.DEPT\_NO

where dept.Loc='SEVILLA';

-- Por supuesto podemos combinar esto junto a

-- funciones de agrupación

-- Queremos mostrar el número de personas por cada número de departamento.

select count(\*) as PERSONAS, dept\_no from EMP

group by dept\_NO;

-- queremos mostrar el número de personas por cada nombre de departamento

select count(\*) as PERSONAS, dept.dnombre

from EMP

Inner join dept

on emp.dept\_no=dept.dept\_no

group by dept.dnombre;

-- Mostrar el Apellido, función y NOMBRE DE HOSPITAL

-- de las personas de la plantilla

select hospital.nombre, plantilla.apellido, plantilla.funcion

from plantilla

inner join hospital

on plantilla.hospital\_cod= hospital.hospital\_cod

order by 1,3;

/\* Tenemos dentro de las consultas de combinación

otro tipo dependiendo de lo que queramos recuperar.

INNER JOIN : recupera los datos que combinen entre las dos tabals.

LEFT JOIN : recupera los datos que combinen y también los datos que no combinan de la tabla de la izqda.

RIGHT JOIN: lo mismo pero de la derecha

FULL JOIN: Recupera todos los datos, combinen o no.

CROSS JOIN: Recupera el producto cartesiano, la combinación de cada fila de una tabla

con las diferentes posibilidades de cada fila de la otra tabla

\*/

select \* from dept;

select distinct dept\_no from emp;

/\* en todas las bases de datos relacionales, existe algo llamado integridad relacional

que hace que los datos estén correctos y que todo esté en orden.

Normalmente, en dicha integridad, lo que hacemos es crear unas NORMAS para que todos

los datos estén relacionados entre sí.

Actualmente, tenemos un Departamento sin EMPLEADOS (40, PRODUCCION, GRANADA)

\*/

-- Vamos a insertar un empleado que no tenga departamento

-- 111, GETAFE, ESTUDIANTE, 1, FECHA, 1, 1, 50

-- Lo metemos como nuevo valor en la tabla de forma manual o

-- INSERT INTO "SYSTEM"."EMP" (EMP\_NO, APELLIDO, OFICIO, DIR, SALARIO, COMISION, DEPT\_NO) VALUES ('111', 'Getafe', 'estudiante', '1', '1240000', '0', '50')

-- la sintaxis del left join o right join no cambia

-- solo tenemos que saber

-- TABLA IZQUIERDA (LEFT): Antes del Join

-- TABLA DERECHA (RIGHT): Después del Join

select emp.apellido, emp.oficio, dept.Dnombre, dept.loc

from EMP

left join DEPT

on EMP.DEPT\_NO = DEPT.DEPT\_NO;

-- Con right join, veremos los departamentos aunque no tengan empleados, es decir, granada

select emp.apellido, emp.oficio, dept.Dnombre, dept.loc

from EMP

right join DEPT

on EMP.DEPT\_NO = DEPT.DEPT\_NO;

-- Full join, muestran todos los datos

select emp.apellido, emp.oficio, dept.Dnombre, dept.loc

from EMP

full join DEPT

on EMP.DEPT\_NO = DEPT.DEPT\_NO;

-- cross join muestra el producto cartesiano (por ejemplo si quiero sacar un coche en cada color

-- tabla coches, tabla colores)

-- no se suele utilizar

-- no lleva ON

select emp.apellido, emp.oficio, dept.Dnombre, dept.loc

from EMP

cross join DEPT;

-- EJERCICIO

--Quiero visualizar el Apellido, Salario y Especialidad junto a la dirección

-- del hospital donde trabaja de todos los doctores de la paz

select d.apellido, d.salario, d.especialidad, h.direccion

from doctor d

inner join hospital h

on d.hospital\_cod=h.hospital\_cod

where h.nombre ='la paz';

-- Quiero visualizar cuántas empleados trabajan en cada departamento

-- mostrando el nombre del departamento

select count(emp\_no) as EMPLEADOS, d.dnombre

from emp e

right join dept d

on d.dept\_no=e.dept\_no

group by d.dnombre;