//para ver una tabla en concreto(con datos)-->select \* from nombre\_tabla

//para ver una tabla en concreto(sin datos)-->describe nombre\_tabla

//para ver una base de datos -->show database

//para ver todas las tablas de una base de datos-->show tables (dentro de la database)

OBTENER FECHAS

month()

monthname()

weekday()

**dayofweek()-usar este**

**obt el id de cada jugador,su nombre y apell, puesto,salario pero mostrando los 3 jugadores que cobran mas→**

select id\_jugador, nombre, apellido, puesto, salario from jugador

order by salario desc

limit 3;

**fecha de hoy:** select current\_date()=curdate();

**mes actual:** select month(current\_date());

**dia de la semana :** select weekday(current\_date());

**si tengo 25 años, en que año naci :**select year(current\_date)-25;

**si soy del 2000. cuantos años tengo:current\_date(year)-2000:**select year(current\_date)-2000;

**obt id,nombre,apell,dia de la semana y mes de la fecha del alta**

select id\_jugador, nombre, apellido,

monthname(fecha\_alta),dayofweek(fecha\_alta)

from jugador;

**LOS DATES: “AAAA/MM/DD”**

**obt el idjugador,nombre apellido y puesto de los 5 jugadores más bajos**

select id\_jugador, nombre, apellido, puesto,altura from jugador

order by altura desc

limit 5;

**obt el idjugador,nombre apellido y puesto segun el orden alfabetico del apellido**

select id\_jugador, nombre, apellido, puesto,altura from jugador

order by apellido

limit 5;

desc de Z a A

select id\_jugador, nombre, apellido, puesto, id\_capitan, fecha\_alta, salario, id\_equipo, altura from jugador;

**PARA CADA JUGADOR MOSTRAR SU NOMBRE Y APELLIDO Y EL NUMERO DE DIAS QUE HAN PASADO DESDE QUE FUERON CONTRATADOS HASTA HOY**

**utilizar el datediff(fecha1,fecha2)**

select nombre,apellido,datediff(current\_date,fecha\_alta) from jugador (para ordenarlos seria **order by 1 (desc)**)

date\_format(“2017-06-15”,”%Y”);

**obt la fecha en :nombre del dia de <dia del mes> de <mes> de <año>**

select date\_format(current\_date(),”La fecha es:%W, %d de %M de %Y”);**PROBAR**

**VACIAR UNA TABLA DE DATOS**->truncate nombre\_tabla; APUNTAR

set foreign\_key\_checks=0;

meto los datos

set foreign\_key\_checks=1;

**SIRVE PARA METER LOS DATOS SIN IMPORTAR SI HACE REFERENCIA A UNA FORANEA QUE TDV NO HE METIDO**

**Lista los partidos que solo se jugaron en noviembre**

select \* from partido **where** month(fecha)=11;

**Partidos que NO se jugaron en enero**

select \* from partido where month(fecha)!=1;

**Los jugadores que no tengan incorporada la altura**

select \* from partido where altura is null;

**Los jugadores que SI tengan incorporada la altura y cobren más de 99000 euros**

select \* from jugador where altura is not null and salario>99000;

**Jugadores que juegan de escolta y pivot(primero que aparezcan los escolta)**

select \* from jugador where puesto=”Escolta” **or** puesto=”Pivot” order by puesto desc; (**No me sale)PREGUNTAR→ME DA ERROR SIEMPRE QUE COPIO Y PEGO→ESCRIBIRLO A MANO EN SQL Y DESPUES SHIFT+CRTL EN EL DOCUMENTO**

**Jugadores cuyo capitán sea 4,6, o 13**

select \* from jugador where id\_capitan=4 or id\_capitan=6 or id\_capitan=13;

**Jugadores que tengan una v en su apellido**

select \* from jugador where apellido like “%v%”;

si empezasen por v-> “%v”;

si terminase-> “v%”;

**NO distingue mayuscula y minuscula**

**Jugadores cuyo nombre termine en “o” o “a”**

select \* from jugador where nombre like “%o” or nombre like “%a”;

**Cuenta el numero de jugadores inscritos**

select count(\*) from jugador;

**Muestra el salario medio de los jugadores**

select sum(salario\_bruto)/count(\* )from jugador; **o** select avg(salario\_bruto) from jugador;

**Cuantos jugadores hay inscritos en cada posicion de juego**

select count(\*), puesto from jugador group by puesto order by 1 desc;

**el 1 es porque del orden depende de la columna 1 (count)**

**Salario medio de los jugadores por puesto**

select avg(salario), puesto from jugador group by puesto order by 1 desc;

**EJERCICIO CONSULTAS HOSPITALES**

**1) Mostrar todos los datos de todas las personas.**

select \* from empleado;

**2) Obtén el DNI, apellidos y función de todas las personas.**

select dni,nombre,funcion from empleado;

**3) Mostrar los apellidos de las personas que vivan en LORCA.**

select nombre from empleado where localidad="LORCA";

**4) Mostrar los apellidos de las personas que vivan en MURCIA o LORCA**

select nombre from empleado where localidad="MURCIA" or localidad="LORCA";

**5) Seleccionar los datos de aquellas personas que vivan en MURCIA y tengan un**

**salario superior a los 1500 euros.**

select \* from empleado where localidad="MURCIA" and salario>1500;

**6) Mostrar los datos de las personas que vivan en MURCIA, tengan un salario superior**

**a los 1500 euros y sean DIRECTORES.**

select \* from empleado where localidad="MURCIA" and salario>1500 and funcion=”DIRECTOR”;

**7) Mostrar los datos de las personas cuya función sea MÉDICO ordenados por**

**apellidos descendentemente.**

select \* from empleado where funcion="MEDICO" order by nombre desc;

**8) Mostrar los datos de todas las localidades que hay en la tabla personas sin**

**repeticiones (debes emplear la cláusula DISTINCT)**

select distinct localidad from empleado;

**9) Mostrar los datos de las personas que tengan un salario superior a 1500 euros y**

**sean médicos. Ordenar la salida por salario descendentemente.**

select \* from empleado where salario>1500 and funcion="MEDICO" order by salario desc;

**10) Seleccionar aquellas personas cuyo apellido comience por M.**

select \* from empleado where nombre **like** "M%";

**11) Mostrar los datos de las personas que tengan una M en el apellido y cuya función**

**sea conserje.**

select \* from empleado where funcion="CONSERJE" and nombre like "%M%";

**12) Mostrar aquellas personas que tengan un salario entre 1500 y 2000 euros.**

select \* from empleado where salario between 1500 and 2000;

**13) Seleccionar los datos de aquellas personas cuya función sea MÉDICO o DIRECTOR**

**(utilizar el operador IN)**

select \* from empleado where funcion **in**("MEDICO","DIRECTOR");

**14) Obtener los datos de aquellas personas cuya función no sea CONSERJE (utilizar el**

**operador NOT IN) y tengan un salario superior a los 1500 euros, ordenados por**

**apellido descendentemente.**

select \* from where funcion not in("CONSERJE") and salario>1500 order by nombre desc; **o** select \* from where not funcion=”CONSERJE” and salario>1500 order by nombre desc;

**15) Mostrar los datos de las personas que sean de MURCIA o CARTAGENA y que**

**pertenezcan al hospital número 1.**

**16) Obtén los apellidos en mayúsculas de las personas que trabajan en el hospital**

**número 1.**

select upper(nombre) from empleado where cod\_hospital=1;

**17) Con una consulta devuelve los apellidos de todas las personas. Al lado debe**

**aparecer la longitud de cada apellido.**

select apellido,char\_length(apellido) from empleados;

**18) Obtener los apellidos y localidad en minúscula de todas aquellas personas que no**

**trabajen en el hospital número 1.**

select lower(apellido),lower(localidad) from empleados where num\_hospital!=1;

**19) Obtener los datos de las personas que trabajan en los hospitales 1 ó 2 y tengan un**

**salario superior a 1500 euros.**

**20) Visualizar los datos de aquellas personas que no trabajen en el hospital número 2 y**

**que sean de MURCIA.**

**Mostrar los apellidos de los empleados cuyo oficio no sea ni conserje ni**

**medico, y además tengan un salario mayor de 1500.**

select \* from where funcion not in (“CONSERJE”,MÉDICO”) and salario>1500;

OTRO EJERCICIO

-- **28) Mostrar los apellidos de los empleados cuyo oficio no sea ni 'ANALISTA' ni**

**'EMPLEADO', y además tengan un salario mayor de 2000.**

select apellido from empleados where oficio not in (“ANALISTA”,”EMPLEADO”) and salario>2000;

-**- 29) Seleccionar de la tabla EMPLE los empleados cuyo salario esté entre 2000 y 3000**

**(utilizar BETWEEN).**

select \* from empleados where salario between 2000 and 3000;

-- **30) Seleccionar el apellido, salario y número de departamento de los empleados cuyo**

**salario sea mayor que 2000 en los departamentos 10 ó 30.**

select apellido,salario, num\_departamento from empleados where salario>2000 and num\_departamento=30 or num\_departamento=10;

**-- 31) Mostrar el apellido y número de los empleados cuyo salario no esté entre 1000 y 2000**

**(utilizar BETWEEN).**

select apellido,count(\*),salario from empleados where salario not between 2000 and 3000 group by salario desc;

-- **32) Obtener el apellidos de todos los empleados en minúscula**

select lower(apellido) from empleados;

-- **33) En una consulta concatena el apellido de cada empleado con su oficio.**

select concat (apellido,oficio) from empleados;

**-- 34) Mostrar el apellido y la longitud del apellido (función LENGTH) de todos los**

**empleados, ordenados por la longitud de los apellidos de los empleados**

**descendentemente.**

select apellido, char\_length(apellido) from empleados order by 2 desc;**(es como count)**

**-- 35) Obtener el año de contratación de todos los empleados (función YEAR).**

select apellido,year(fecha\_alta) from empleados;

**-- 36) Mostrar los datos de los empleados que hayan sido contratados en el año 1992.**

select \* from empleados where year(fecha\_alta)=1992;

**-- 37) Mostrar los datos de los empleados que hayan sido contratados en el mes de febrero**

**de cualquier año (función MONTHNAME).**

select \* from empleados where month(fecha\_alta)=2;

-- **38) Para cada empleado mostrar el apellido y el mayor valor del salario y la comisión que**

**tienen (funciones GREATEST y COALESCE)**

select apellido,greatest(salario),coalesce(comision) from empleados;

**-- 39) Mostrar los datos de los empleados cuyo apellido empiece por 'A' y hayan sido**

**contratados en el año 1990.**

select \* from empleados where apellido like “A%” and year(fecha\_alta)=”1990”;

**-- 40) Mostrar los datos de los empleados del departamento 10 que no tengan comisión.**

select \* from empleados where num\_departamento=10 and comision is null;

**AS**

select salario as sueldo from empleado;

ALTER TABLE ENTIDAD ADD FOREIGN KEY() REFERENCES ENTIDAD2 ();

DROP TABLE NOMBRE;TRUNCATE TABLE NOMBRE;DESCRIBE NOMBRE

EJERCICIO

30->Nombre de las asignaturas impartidas por profesores de Logroño

31->nombre completo alumnos

32->DNI, nombre y apellidos de los alumnos que viven en La Rioja.Como criterio usar el prefijo 491

33->DNI, nombre y apellido de los alumnos matriculados en la asignatura Seguridad Vial

34->ID de las titulaciones en la que está matriculado el alumno con DNI 20202020A.Un alumno está matriculado en una titulación si está matriculado en una asignatura de la titulación

35-> Nombre de las asignaturas en la que está matriculada Rosa Garcia,mostrando tmb por cada asignatura el número de matrícula

36->Dni de los alumnos a los que imparte clase el profesor Jorge Saez

37->Cuál sería el coste global de cursar la titulación de Matemáticas si el coste de cada asignatura fuera incrementado un 7%?

**2/2**

**marca de coche que tenga más modelos**

marca con mas modelos

select marca,count(modelo)

from coches

group by marca

having count(modelo)=(select max(total)

from (select marca,count(modelo) as total

from coches

group by marca) as coches\_total);

/\*nombre del autor que mas libros ha escrito\*/

select libros.autor,count(libros.codigo\_libro)

from libros

group by libros.autor

having count(libros.codigo\_libro)=(select max(total)

from(select libros.autor,count(libros.codigo\_libro) as total

from libros

group by libros.autor) as libros\_total);