

Mayo Semana 3

💡 13/5/2024

mysql ~u root ~p <contraseña>

contraseña :novauniversitas

💡 14/5/2024

show databases; //acceder a la base de datos

use world; //Estoy en tupla world

use sakila; //Salgo de world y me voy a sakila

show tables; //mostrar tablas

create database prueba; //crear base de datos llamada prueba

drop database prueba; //eliminar base de datos llamada prueba

💡 15/5/2024

create table emp (nd varchar(2),nomd varchar(25),dnisuper varchar(3)); //crear una tabla, en los parentesis va el nombre y tipo de dato, tamaño maximo que se pueden meter en el atributo

- Tambien se puede ir ingresando columna por columna dando un enter por cada columna que haga

show columns from dpto; //Ver las columnas de la tabla

💡 16/5/2024

No class

💡 17/5/2024

nomp varchar(20) unique //Hace unico un atributo no se puede repetir

pais varchar(15) not null //Hace que un atributo no se quede con valor nulo

primary key(np) // indicando cual va a ser la llaver primaria

hora integer check(hora>0) //validar algun tipo de dato

alter table dpto add primary key(nd); //alterar tabla para decirle cual es el atributo que sera llaver primaria

- add → agregar

alter table dpto //modificar un atributo para que sea llave foranea

add foreign key(dnisuper) references emp(dni);

alter table emp //agregar atributo -> add sexo varchar(1);

alter table emp // eliminar atributo -> drop sexo;

alter table emp //Cambiar tamaño y nombre -> change sexo sex varchar(1);

Mayo Semana 4

💡 20/5/2024

alter table dpto modify nomd varchar(15) //modificando tamaño de nomd a 15 caracteres

drop table dpto; //Borrar una tabla

truncate table emp; //Elimina datos de la tabla,

insert into proy(np, nomp, pais,nd)

values ('P1', 'TELECONFERENCIA','ESPAÑA','D1');

select * from proy; //seleccionar filas

💡 21/5/2024

//copiando a otra tabla, se puede llenar solo una columna

insert into dpto2 (nd, nomd, dnisuper)

→ select nd,nomd,dnisuper

→ from dpto;

//Usando condicionales

insert into dpto2(nd,nomd,dnisuper) -> select * -> from dpto -> where nd = 'D2'; //seleccionar registros de dpto donde nd = 'D2'

<<where nd= 'D1' or nd='D4'; //seleccionando usando dos condicionescolo

update dpto //actualizar valores de una tabla

→ set nomd = 'RH'

→ where nd = 'D2' //cambiando nomd donde nd sea igual a 'D2'

insert into dpto (nd, nomd, dnisuper) -> values ('D1','INVESTIGACION','E8'), -> ('D2','RECURSOS H','E6'), -> ('D3','ADMINISTRACION','E4'), -> ('D4','DESARROLLO SW','E11'), -> ('D5','PRUEBAS','E12'), -> ('D6','ANALISIS','E13'), -> ('D7','DISEÑO INT','E10');

💡 22/5/2024

delete from dpto

→ where nd = 'D7'; //Borrar registro de la tabla usando nd para especificar cual es el que va a ser eliminado

INSTRUCCIONES DE BUSQUEDA

💡 23/5/2024

Hacer bases de datos DROP

💡 24/5/2024

Hacer base de datos de los libros

Mayo Semana 5

💡 27/5/2024

Respalidar base de datos

C:\Users\Admin\Documents\dumps\Dump20240527.sql

💡 28/5/2024

Comando select

select * //que aparezcan todos

→ from proy // de que tabla va a sacar la informacion que estoy seleccionando

from proy, trabaja_en, dpto //Producto cartesiano

→ where //indicar a la seleccion la condicion

→ np = "P1" and np = "P4" //condicion de seleccion

select nomp, pais //seleccionando de proy solo el -> from proy; //nombre y los paises

//seleccionar las filas correctas despues del producto cartesiano

select *

from proy, emp

where proy.nd = emp.nd

//seleccionar nombre de emp, despues del producto castesiano emp x proy, filtrando la informacion correcta de emp x proy y seleccionando quienes trabajen en nutrimex

select emp.nomp -> from emp, proy -> where emp.nd=proy.nd and proy.nomp = "NUTRIMEX";

//seleccionar el nombre del departamento donde trabaja Lidia

select dpto.nomd -> from emp, dpto -> where emp.nd=dpto.nd and emp.nomp="LIDIA" and emp.ap1="RUIZ" and emp.ap2="MENDOZA";

//nombre y apellido de los empleados que ganan mas o igual a 7000 pesos

select emp.nomp, emp.ap1 -> from emp -> where emp.sueldo>=7000;

💡 29/5/2024

No class

💡 30/5/2024

No class

💡 31/5/2024

No class

Junio Semana 1,2

💡 3/6/2024

funciones count y sum

- Ejemplos de consultas de clase

```
select count(nomp) //contar numero de personas que cumplen con la condicion -> from emp ->
where sueldo <= 10000;
```

```
//seleccionar y agrupar
```

```
select nomp, count(nomp) from emp where sueldo >= 10000 group by nomp;
```

```
//usando atributo dni
```

```
select dni, sum(hora) from trabaja_en where hora >5 //el where es opcional group by dni;
```

```
//atributo para agrupar
```

```
//
```

```
select dni,sum(hora) -> from trabaja_en -> group by dni;
```

```
//
```

```
select np, sum(hora) -> from trabaja_en -> group by np;
```

```
//
```

```
select proy.np, sum(hora) -> from trabaja_en, proy -> where trabaja_en.np=proy.np -> group by
proy.np;
```

```
//
```

```
select proy.nomp, sum(hora) -> from trabaja_en, proy -> where trabaja_en.np=proy.np -> group
by proy.np;
```

```
//
```

```
select max(sueldo) -> from emp -> group by sueldo;
```

```
//Seleccionar el sueldo maximo de la tabla
```

```
select max(sueldo) -> from emp;
```

```
//Seleccionar el nombre de la persona con el sueldo maximo
```

```
select nomp -> from emp -> where sueldo = (select max(sueldo) from emp);
```

//Renombramiento

```
select * -> where sueldo = (select max(sueldo) from emp) as c1;
```

💡 4/6/2024

having, order by, modificador in, between, like "a%"

- ejemplos

//Seleccionar dni, agrupar por horas trabajadas si la suma de las horas es mayor a 40

```
select dni, sum(hora) -> from trabaja_en -> group by dni -> having sum(hora) > 40;
```

//Renombramiento de la agrupacion de horas a horas trabajadas

```
select dni, sum(hora) as horas_trabajadas -> from trabaja_en -> group by dni -> having  
horas_trabajadas > 40;
```

//Ver la tabla por un orden especifico asc: ascendente. desc: descendente

```
select * -> from emp -> order by sueldo desc;
```

//visualizar solo los nombres ordenados de acuerdo a su sueldo

```
select nomp -> from emp -> order by sueldo desc;
```

//Ordenar por orden alfabetico

```
select nomp -> from emp -> order by nomp asc;
```

//ordenar el nombre de los empleados que ganan mas de 7000

```
select nomp -> from emp -> where sueldo > 7000 -> order by nomp asc;
```

//Otra forma de seleccionar en lugar de usar where largo

```
select nomp, nd -> from emp -> where nd in ("D1","D2","D3");
```

//Seleccionar el nombre y sueldo de la persona que gana entre 7000 y 12000

```
select nomp, sueldo -> from emp -> where sueldo between 7000 and 12000;
```

//seleccionar de acuerdo a las letras de una cadena

"a_d": que tengan a al principio y d al final, "al%": que contengan al en cualquier parte, depende donde esta el porcentaje se tomara en cuenta

```
select nomp -> from emp -> where nomp like "%u%";
```

5/6/2024

- inner (union)

-inner join(seleccionan los que tienen en comun)

//

select * -> from emp inner join dpto on emp.nd=dpto.nd;

//Producto cartesiano sin restriccion

select * -> from emp cross join dpto;

- union por izquierda y derecha

por izquierda

//—

select * -> from emp left join dpto on emp.nd=dpto.nd;

//—Lo que no coincide lo pone null

select * -> from emp left join dpto on emp.dnisuper=dpto.dnisuper;

//—A departamentos le pegamos empleados

select * -> from emp right join dpto on emp.dnisuper=dpto.dnisuper;

- union, une resultado de dos consultas

//—

select dpto.nd

from dpto

//—unir estas consultas

select proy.nd

from proy

//—

select dpto.nd -> from dpto -> union -> select proy.nd -> from proy;

//—saber que apellidos tenemos en la base de datos

select emp.ap1 -> from emp -> union -> select emp.ap2 -> from emp;

//—ordenar los apellidos que se tienen en la base de datos

select emp.ap1 -> from emp -> union -> select emp.ap2 -> from emp -> order by ap1 asc;

💡 6/6/2024

- Consultas 2
- 1.Cuantas horas trabaja cada empleado, ordenar de menor a mayor

```
select emp.nomp, trabaja_en.hora -> from emp, trabaja_en -> where emp.dni = trabaja_en.dni -  
> order by trabaja_en.hora asc;
```

2.Cual es el promedio de horas trabajadas por departamento

- 3.Cuantos trabajadores son hombres y cuantos son mujeres

```
alter table emp -> add sexo varchar(1);
```

```
update emp set sexo = "M" where nomp="ABELARDO";
```

```
update emp set sexo = "F" where nomp="LIDIA";
```

```
update emp -> set sexo = "M" -> where nomp="JUANCARLOS";
```

```
update emp set sexo = "F" where nomp="SONIA";
```

```
update emp -> set sexo = "M" -> where nomp="MANUEL";
```

```
update emp -> set sexo = "M" -> where nomp="ALBERTO";
```

```
update emp -> set sexo = "M" -> where nomp="IVAN";
```

```
update emp -> set sexo = "M" -> where nomp="RODRIGO";
```

```
update emp -> set sexo = "M" -> where nomp="ALAN";
```

```
update emp -> set sexo = "M" -> where nomp="ADRIAN";
```

```
update emp -> set sexo = "F" -> where nomp="CARLOTA";
```

```
update emp -> set sexo = "F" -> where nomp="CAMILA";
```

```
update emp -> set sexo = "F" -> where nomp="DIANA";
```

```
update emp -> set sexo = "M" -> where nomp="DIEGO";-
```

```
update emp -> set sexo = "M" -> where nomp="JUAN";
```

[4.De](#) acuerdo a la pregunta anterior, hay mas empleados hombres o mujeres?

5.Quien es el supervisor que trabaja mas tiempo?

6.Cuanto gana cada empleado, si ahora se paga 200 pesos cada hora

[7.De](#) la pregunta anterior cual es el total que se debe pagar?

8.Quienes son los empleados que no trabajan en el departamento de administración o diseño de interfaces

9.Que proyecto trabaja mas horas?

10.En que país se trabaja menos horas

Hacer respaldo(exportar) de base de datos a modo comando:mysqlDump --user=root -p prueba > respaldoPrueba.sql

//Hacer lo contrario, importar

```
mysql --user=root -p prueba < C:\Users\Admin\respaldoPrueba.sql
```