

Nombre del Trabajo:

Practica 2 Unidad 2(HOSPITAL)



Saldaña Aguilar Gabriela

Materia BASES DE DATOS

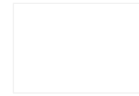
Profesor HERNANDEZ CONTRERAS EULER

Grupo 2CM10

Fecha de entrega 19/01/15

INDICE.....

1.-Marco Teórico.....3



Desarrollo.....3

2.-CONCLUSIONES.....5

3.-BIBLIOGRAFÍA.....5

MARCO TEÓRICO

MySQL Workbench es una herramienta CASE visual unificada para los arquitectos de bases de datos, desarrolladores y administradores de bases. MySQL Workbench proporciona el modelado de datos, desarrollo de SQL y herramientas completas de administración de la configuración del servidor, la administración de usuarios, copia de seguridad, y mucho más.

Modelar simplifica el diseño de bases de datos y mantenimiento, ya que permite, el arquitecto de datos, para visualizar los requisitos y resolver problemas de diseño. Diseño de base de datos basada en modelos es una metodología eficiente para la creación de bases de datos válidos y de buen rendimiento, mientras que proporciona la flexibilidad necesaria para responder a las cambiantes necesidades de datos. Los modelos se utilizan para construir los diagramas ER y bases de datos MySQL físicas.

En esta práctica se continuo con el tema ya visto de PROYECCIONES aumentando sintaxis nueva como fue DISTINCT y BETWEEN los cuales no habíamos visto en clases anteriores, con esta práctica se busca reforzar la habilidad para generar búsquedas.

TAREAS A REALIZAR:

- 1.-describir cada tabla
- 2.-sacar el modelo relacional
- 3.- **MOSTRAR LOS DATOS CORRESPONDIENTES A LOS PACIENTES**

```
select * from paciente  
order by edad;
```

4.-MOSTRAR EL NOMBRE Y LA EDAD DE LOS PACIENTES

```
select nombre as NOMBRE, edad as EDAD
from paciente;
```

5.-MOSTRAR LOS DATOS DEL PACIENTE CUYO CURP SEA "MALD770810"

```
select * from paciente
where curp like "MALD770810";//como dan una cadena de caracteres
conviene usar like
```

6.-MOSTRAR LOS DATOS DE LOS PACIENTES QUE TIENEN 26 AÑOS.

```
select * from paciente
where edad=26// se pueden comparar ya que ambos son tipo int
order by nombre;
```

7.-MOSTRAR LOS DATOS DE LOS PACIENTES CUYA EDAD SEA MAYOR A 25 AÑOS.

```
select * from paciente
where edad>25// esto se puede realizar entre el mismo tipo
order by nombre;
```

8.-MOSTRAR LOS PACIENTES CUYA EDAD SEA MAYOR O IGUAL A 27 AÑOS.

```
select * from paciente
where edad>=27
order by nombre;
```

9.-MOSTRAR LOS PACIENTES CUYA EDAD SEA MENOR O IGUAL A 27 AÑOS.

```
select DISTINCT * from paciente
where edad<=27
order by nombre;
```

10.-MOSTRAR LOS PACIENTES CUYA EDAD SEAN IGUAL A 26,27 Y 28 AÑOS.

```
select * from paciente
where edad in(26,27,28)--in se usa para elegir varios valores
dentro de los que queremos buscar
order by nombre;
```

11.-MOSTRAR LOS PACIENTES CUYA EDAD SEAN IGUAL A 26 Y 28 AÑOS.

```
select * from paciente
where edad in(26,28)
order by nombre;
```

```
select * from paciente
where edad >=26
and edad not in(28)//hace la misma función que in pero negando el
resultado
order by nombre;
```

```
select * from historial
where fechaIngreso="2003-03-26">//ambos son caracteres tipo
date(año,mes,dia) por eso van entre comillas
order by idHistorial;
```

```
select * from historial
where fechaIngreso>"2003-03-25";//se pueden usar los operadores
>,<,>=,<=,=,!=
```

```
select * from historial
where fechaIngreso between "2003-03-25"--valor inicial
and "2003-03-27" ; --valor final
```

```
select count(*) from historial//trae el total
where idCausaHosp=2;
```

En esta práctica pude realizar mejor y con menos errores mis búsquedas a pesar de la sencillez de estas, aprendí nueva sintaxis que puede ayudar a realizar una búsqueda más fácilmente y sin tantas líneas de código.

Bibliografia: PROFESOR EULER.