

Guía de Consultas (Simusaber 3 y 5)

VERIFICAR LAS TABLAS DE LA BASE DE DATOS

Show tables;

SELECCIÓN DE TABLA PRUEBA

Select * from prueba;

SELECCIÓN DE TABLA USUARIOS

Select * from usuarios;

CONSULTA ENTRE DOS TABLAS (USUARIOS Y PRUEBA), SE QUIERE VERIFICAR QUE TIPO DE PRUEBA REALIZO EL ESTUDIANTE

Select

usuarios.nombres as nombre,

prueba.tipo_prueba as tipo,

prueba.descripcion_prueba as descripcion

from prueba

inner join usuarios on prueba.id_usuario = usuarios.id_usuario;

CONSULTA DE TRES TABLAS

Select * from usuarios;

Select * from rol;

Select * from prueba;

CONSULTA ENTRE TRES TABLAS (USUARIOS, ROL Y PRUEBA), SE BUSCA VERIFICAR EL USUARIO QUE REALIZO LA PRUEBA QUE TIPO DE ROL TIENE Y ADICIONAL LA FECHA EN QUE LA APLICO

select

usuarios.id_usuario as id_usuario,

```
usuarios.nombres as nombre,  
rol.rol as rol_usuario,  
prueba.id_prueba as id_prueba,  
prueba.fecha_de_aplicacion as fecha_aplicacion  
from prueba  
inner join usuarios on prueba.id_usuario = usuarios.id_usuario  
inner join rol on rol.id_usuario = usuarios.id_usuario;
```

CONSULTA DE DOS TABLAS

```
Select * from usuarios;
```

```
SELECT * from sala;
```

CONSULTA ENTRE DOS TABLAS, PARA VERIFICAR CUALES USUARIOS TIENEN SALAS CREADAS

```
select  
usuarios.id_usuario as id_usuario,  
usuarios.nombres as nombre,  
usuarios.apellidos as apellido,  
sala.id_sala as id_sala  
from usuarios  
left join sala on usuarios.id_usuario = sala.id_usuario;
```

CONSULTA DE DOS TABLAS

```
Select * from usuarios;
```

```
Select * from prueba;
```

CONSULTA ENTRE DOS TABLAS, PARA VER CUAL FUE EL RESULTADO LOS USUARIOS EN LAS PRUEBAS

```
select
usuarios.id_usuario as id_usuario,
usuarios.nombres as nombre,
usuarios.apellidos as apellido,
prueba.resultado as resultado
from prueba
right join usuarios on prueba.id_usuario = usuarios.id_usuario;
```

CONSULTA PARA VER EL NATURAL JOIN ENTRE DOS TABLAS (SALA Y USUARIOS)

```
select *
from sala
natural join usuarios;
```

CONSULTA PARA VER EL NATURAL JOIN ENTRE DOS TABLAS (BANCO_PREGUNTAS Y ESTRUCTURA_BANCO)

```
select *
from banco_preguntas
natural join estructura_banco;
```

CONSULTA DE TABLA USUARIOS

```
SELECT * from usuarios;
```

INSERTA UN NUEVO USUARIO EN LA TABLA USUARIOS

```
insert into
usuarios (id_usuario, nombres, apellidos, usuario,
correo, contrasena, rol, id_colegio)
values (1031, "Camilo Stiwar", "Lopez Ferreira", "Cafelo",
```

```
"cafelo124@gmail.com", "aw15a2", 2, 1256984);
```

```
SELECT * from usuarios where id_usuario = 1031;
```

INSERTAR UN NUEVO USUARIO EN LA TABLA USUARIOS, SIN TODOS LOS CAMPOS

Insert into

```
usuarios (nombres, rol) values ("Ricardo Estiven", "Richard");
```

```
select * from usuarios;
```

ELIMINAR USUARIO CON DATOS EN ATRIBUTOS INCORRECTOS

```
delete from usuarios where rol = 0;
```

CONSULTA PARA VERIFICAR QUE EL USUARIO FUE ELIMINADO

```
select * from usuarios;
```

INSERTAR NUEVAMENTE EL USUARIO CON LOS CAMPOS CORRECTOS EN LA TABLA USUARIOS

Insert into

```
usuarios (nombres, usuario, rol) values ("Ricardo Estiven", "Richard", 2);
```

```
select * from usuarios where nombres ="Ricardo Estiven";
```

ACTUALIZAR LOS DATOS DEL USUARIO QUE SE ENCONTRABAN INCOMPLETOS A LA HORA DE REGISTRARLO EN LA TABLA USUARIOS

update usuarios

```
set id_usuario = 1032, apellidos = "Arjona Cepeda",
```

```
correo = "ricardoarjo@hotmail.com",
```

```
contrasena = "45eltaxista",
```

id_colegio = 8521478

where nombres = "Ricardo Estiven";

VERIFICAR SI SE ENCUENTRA ACTUALIZADO

select * from usuarios where nombres ="Ricardo Estiven";

USUARIO CON CORREO DESACTUALIZADO

select * from usuarios where usuario = "pauli";

ACTUALIZAR CORREO DE USUARIO EN LA TABLA USUARIOS

update usuarios

set correo = "pauli1996@gmail.com"

where usuario = "pauli";

select * from usuarios where usuario = "pauli";

INSERTAR UN USUARIO REPETIDO APLICANDO EL COMANDO IGNORAR PARA QUE LO DEJE AGREGAR A LA TABLA USUARIOS

Insert ignore into

usuarios (nombres, usuario, rol) values ("Ricardo Estiven", "Richard", 2);

SE ELIMINA EL USUARIO REPETIDO DE LA TABLA

delete from usuarios where id_usuario = 0;

select * from usuarios where usuario = "Cafelo";

ACTUALIZAR EL ROL DEL USUARIO CAFELO, YA QUE TENIA ROL 2 DE ESTUDIANTE Y ES DOCENTE CON ROL 3

update

usuarios set rol = 3

where usuario = "Cafelo";

VERIFICAR SI EL ROL SE ACTULIZO EN LA TABLA

```
select * from usuarios where usuario = "Cafelo";
```

ACTUALIZAR DATOS EN DOS TABLAS (USUARIO Y SALA)

```
update  
sala sal join usuarios usu  
on (sal.id_sala = usu.id_usuario)  
set  
sal.id_sala = 18,  
usu.id_usuario = 1034  
where  
usu.usuario = "Cafelo" and sal.id_sala is null;
```

SELECCIÓN DE TABLA COLEGIOS

```
select * from colegios;
```

INSERTAR NUEVO COLEGIO EN LA TABLA COLEGIOS

```
insert into  
colegios (id_colegio, nom_colegio, id_usuario, id_sala)  
values (112233, "Uniminuto", 1020, 22);
```

```
select * from colegios where id_colegio = 112233;
```

ELIMINAR DATO DE COLEGIO REGISTRADO POR INFORMACIÓN INCORRECTA

```
delete from colegios where id_Sala = 22;
```

```
select * from colegios;
```

INSERTAR DE NUEVAMENTE EL COLEGIO EN LA TABLA COLEGIOS

```
insert into  
colegios (id_colegio, nom_colegio, id_usuario, id_sala)  
values (112233, "Uniminuto", 1020, 25122);
```

SELECCIÓN DEL DATO AGREGADO

```
select * from colegios where id_colegio = 112233;
```

ACTUALIZAR INFORMACIÓN DEL COLEGIO AGRADO, SU ID ESTABA INCORRECTO

```
update colegios  
set id_colegio = 112323  
where nom_colegio = "Uniminuto";
```

VERIFICAR ACTULIZACIÓN DE DATO

```
select * from colegios where id_colegio = 112323;
```

SELECCIÓN DE TABLA PRUEBA

```
select * from prueba;
```

APLICACIÓN DE CASE, BUSCAMOS VERIFICAR QUE USUARIOS APROBARON Y NO APROBARON CON BASE AL RESULTADO OBTENIDO EN SU PRUEBA. MENOR Y MAYOR AL 60%

```
Select  
id_prueba, fecha_de_aplicacion,  
id_usuario,  
case  
when resultado < 60  
then 'Noaprobo'  
when resultado > 61  
then 'Aprobo'
```

```
end Estado_prueba
```

```
from prueba;
```

```
select * from prueba;
```

APLICACIÓN DE CASE, SE BUSCA VERIFICAR SI LA PRUEBA ES DEL AÑO ANTERIOR O DEL RECIENTE

```
select
```

```
id_prueba, fecha_de_aplicacion,
```

```
id_usuario,
```

```
case
```

```
when fecha_de_aplicacion < '2020-12-31'
```

```
then 'pruebaAntigua'
```

```
when fecha_de_aplicacion > '2020-12-31'
```

```
then 'pruebaReciente'
```

```
end tiempo_prueba
```

```
from prueba;
```

SELECCIÓN DE TABLA USUARIOS

```
select * from usuarios;
```

ACTUALIZAR Y CIFRAR CONTRASEÑA DEL USUARIO

```
update usuarios
```

```
set contrasena = aes_encrypt ('1001271', 'secretonacional')
```

```
where id_usuario = 1004;
```

```
select * from usuarios where id_usuario = 1004;
```

SELECCIÓN Y DESCIFRADO DE LA CONTRASEÑA DEL USUARIO


```
select aes_decrypt (contrasena, 'secretonacional')  
from usuarios where id_usuario = 1004;
```

ACTUALIZAR Y COMPRIMIR CONTRASEÑA DEL USUARIO

```
update usuarios  
set contrasena = compress ('waltereselmaslindo..')  
where id_usuario = 1001;
```

```
select * from usuarios;
```

SELECCIONAR Y DESCOMPRIMIR CONTRASEÑA DEL USUARIO

```
select uncompress(contrasena) from usuarios where id_usuario = 1001;
```