

PRÁCTICA 2: BBDD vs sistemas de ficheros (II)

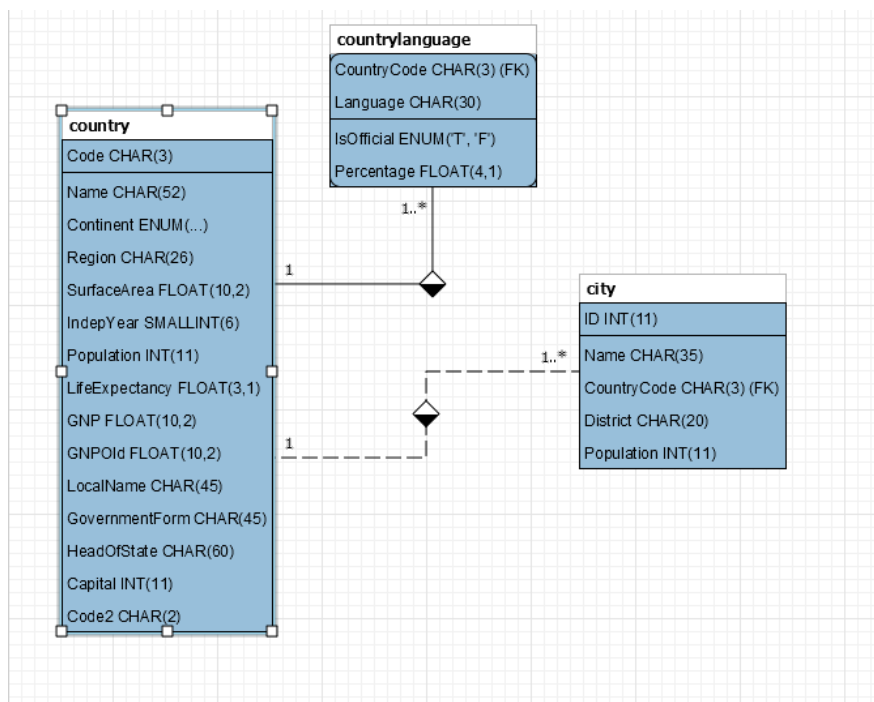
En esta práctica utilizaremos un sistema gestor de bases de datos para:

- Identificar las principales ventajas que nos ofrece un sistema gestor de bases de datos.
- Identificar los lenguajes DML, DDL y DCL.
- Identificar el nivel conceptual y el nivel externo.
- Entender la independencia lógica de la base de datos.
- Observar cómo la base de datos nos permite evitar accesos no autorizados.
- Acceder a una base de datos desde varios ordenadores.
- Gestionar los usuarios de la base de datos.

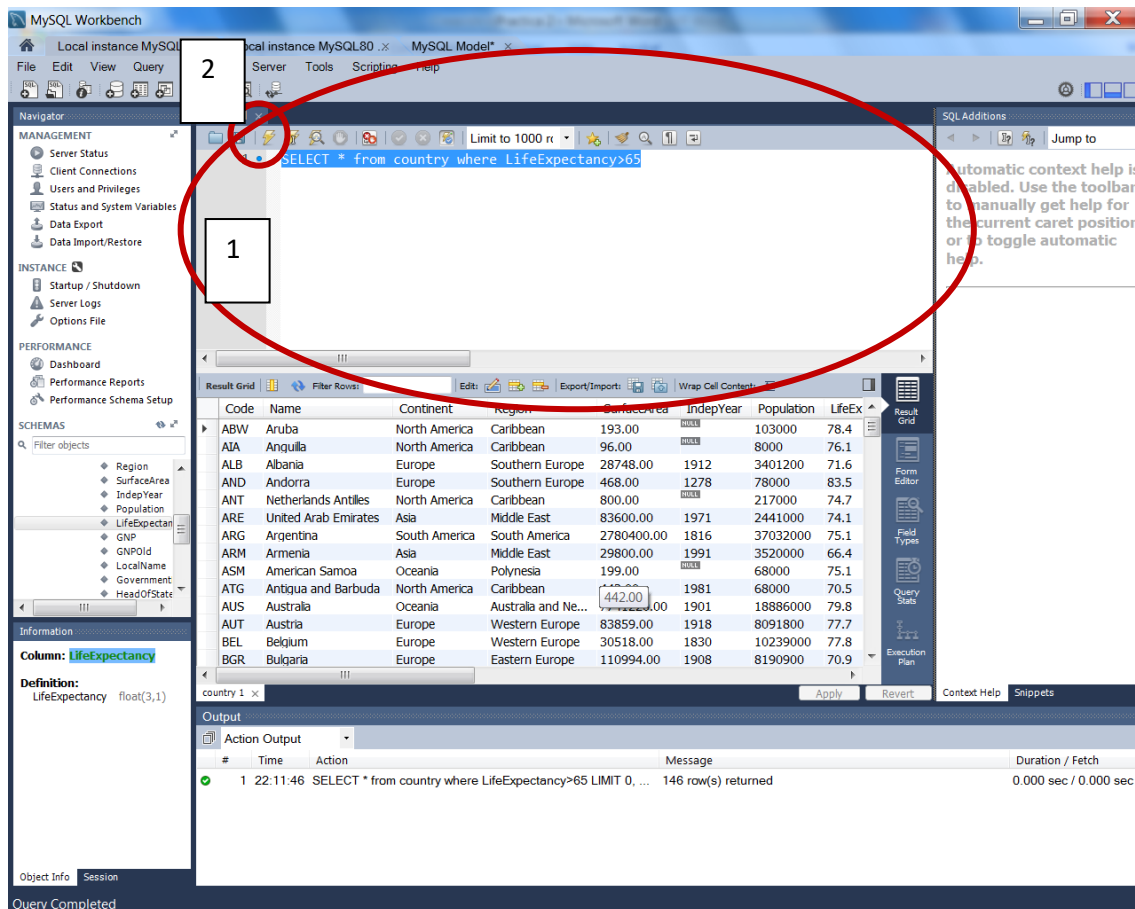
Anotaciones:

- Utilizaremos la BBDD de MySQL “world”.
- El lenguaje SQL se estudiará más adelante por lo que nos limitaremos a comentar los resultados.
- Los ejercicios serán parecidos a los de la práctica 1 para comparar los dos sistemas.

1. Abrir la BBDD “world”. Esta base de datos almacena los datos de todos los países del mundo. Cada país tiene una serie de ciudades de las cuales también almacena sus datos. También almacena todos los idiomas que se habla en cada país. Tenemos el siguiente esquema:



2. Queremos saber todos los países que tienen una esperanza de vida de más de 65 años.
 - a. Ejecutar la consulta: "SELECT * from country where LifeExpectancy>65"
 - b. Para ejecutar pegar el código sobre el área 1 y pulsar el botón 2 (rayo).



- c. Ejecutar también: "SELECT * from country where LifeExpectancy>80" y "SELECT * from country where LifeExpectancy>70 order by name desc".
3. Crear un nuevo campo que nos va a permitir calcular la densidad de población de cada país:
 - a. Crearemos una consulta que haga exactamente lo mismo que la hoja "estadísticas" de la hoja de cálculo.

Select "Estadísticas de paises", "" from dual

union

select "superficie total del mundo", cast(sum(SurfaceArea) as char) from country

union

Select "Países con esperanza de vida mayor de 60 años", cast(count(*) as char) from country where LifeExpectancy >60

union

Select "Países con esperanza de vida mayor de 70 años", cast(count(*) as char) from country
where LifeExpectancy >70

union

Select "Países con esperanza de vida mayor de 75 años", cast(count(*) as char) from country
where LifeExpectancy >75

union

Select "Poblacion total mundial", cast(sum(population) as char) from country

union

select "Países con mas de 30 millones de habitantes", cast(count(*) as char) from country
where population >30000000

- b. Crear una nueva columna en la tabla "country" y llenarla con los valores correspondientes.

Alter table country
add column avgpopulation float(10,2)
after Population

update country set avgpopulation=Population/SurfaceArea

4. Añadir una nueva funcionalidad en la tabla de países (country). Esta nueva funcionalidad nos permitirá saber que montañas hay en cada país:
- Añadir una nueva columna con: "alter table country add column mountain char (60)".
 - Añadiremos 3 nuevas filas con las montañas de España:

```
insert into country values ('ESP', 'Spain', 'Europe', 'Southern Europe', '505992.00',
'1492', '39441700', '77.95', '78.8', '553233.00', '532031.00', 'España', 'Constitutional
Monarchy', 'Juan Carlos I', '653', 'ES', 'Teide');
insert into country values ('ESP', 'Spain', 'Europe', 'Southern Europe', '505992.00',
'1492', '39441700', '77.95', '78.8', '553233.00', '532031.00', 'España', 'Constitutional
Monarchy', 'Juan Carlos I', '653', 'ES', 'Mulacén');
insert into country values ('ESP', 'Spain', 'Europe', 'Southern Europe', '505992.00',
'1492', '39441700', '77.95', '78.8', '553233.00', '532031.00', 'España', 'Constitutional
Monarchy', 'Juan Carlos I', '653', 'ES', 'Aneto');
```

- c. No nos deja. ¿Cuál es el motivo? El SGBD controla que no se introduzcan datos duplicados en la tabla
5. Intentar cambiar la ciudad de París y poner que pertenezca a un código de país que no existe FRN:
- Ejecutar “update city set CountryCode='FRN' where Name='Paris'”
 - ¿Permite el cambio? ¿Por qué?
 - Ejecutar el siguiente código para insertar una nueva ciudad “Carcasone” con un código de país que no existe: “insert into city values(null,'Carcasone','FRN','Carcasona',50000)”
 - El SGBD impide la inconsistencia de datos.
6. Cambiar el código de país de Bélgica, en vez de BEL queremos que aparezca BEG:
- Ejecutar “update country set code='BEG' where code='BEL'”
 - No podemos cambiar el código porque el sistema indica que hay una relación con la tabla “city”: **Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`world`.`city`, CONSTRAINT `city_ibfk_1` FOREIGN KEY (`CountryCode`) REFERENCES `country` (`Code`))**
 - Ejecuta las siguientes sentencias:

```
ALTER TABLE city

DROP FOREIGN KEY `city_ibfk_1`,

ADD CONSTRAINT `city_ibfk_2` FOREIGN KEY (`CountryCode`)

REFERENCES `country` (`Code`)

ON DELETE CASCADE

ON UPDATE CASCADE;
```

```
ALTER TABLE countrylanguage

DROP FOREIGN KEY `countryLanguage_ibfk_1`,

ADD CONSTRAINT `countryLanguage_ibfk_2` FOREIGN KEY (`CountryCode`)

REFERENCES `country` (`Code`)

ON DELETE CASCADE

ON UPDATE CASCADE;
```

```
update country set code='BEG' where code='BEL'
```

- d. Observa qué ha pasado con el país Bélgica mediante las siguientes sentencias:

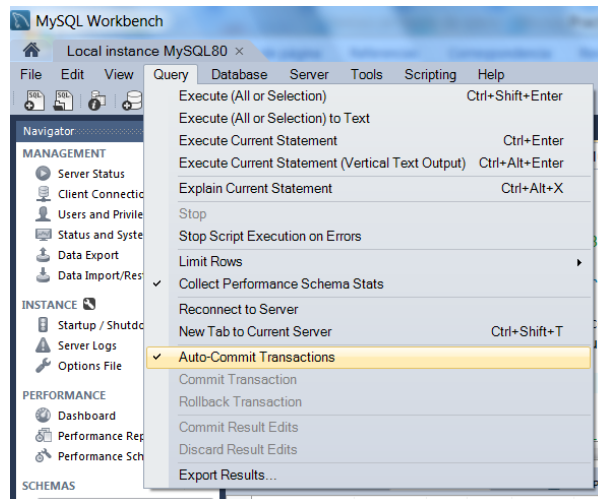
```
select * from country where name='Belgium'
```

```
select * from city where countrycode='BEG'
```

```
select * from countrylanguage where countrycode='BEG'
```

```
select * from city where countrycode='BEG'
```

7. Borrar un país porque se ha fusionado con otro más grande:
- Deshabilitar la opción Autocomit Transaction en el Workbench para que no se guarden las modificaciones inmediatamente en el disco (más info en: [enlace](#)).



- Ejecutar las siguientes sentencias:

```
Delete from country where code=' MNG'
```

```
Select * from country where code='MNG';
```

```
Select * from city where countrycode ='MNG';
```

Select * from countrylanguage where countrycode='MNG';

- c. Vemos que se han borrado los países las ciudades y los idiomas, no es lo que queríamos. Las ciudades se deberían de haber mantenido ya que no han desaparecido, solo se tienen que reasignar a otro país. Esto es debido a un error en el diseño de la base de datos que hemos cometido nosotros en el anterior ejercicio, el cual debemos de corregir. Ejecuta “rollback” y vuelve a ejecutar los “select” del punto anterior. ¿Qué hace la sentencia “rollback”?

8. Ejecuta las siguientes sentencias y sopesa qué información es devuelta:

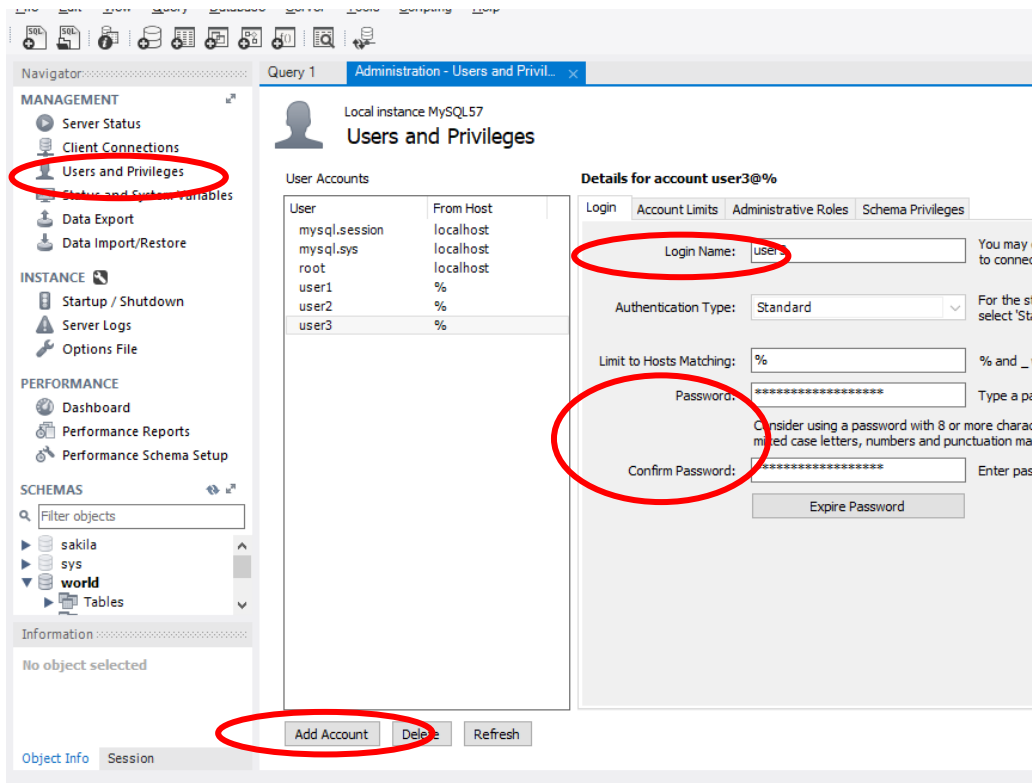
- a. DESCRIBE city;
- b. DESCRIBE country;
- c. DESCRIBE countrylanguage;

9. Vamos a probar a trabajar con los mismos datos, pero con diferentes usuarios. De esta manera podremos observar cómo trabaja la base de datos cuando tiene a varios usuarios conectados y las diferentes maneras que tiene de garantizar la fiabilidad de la información:

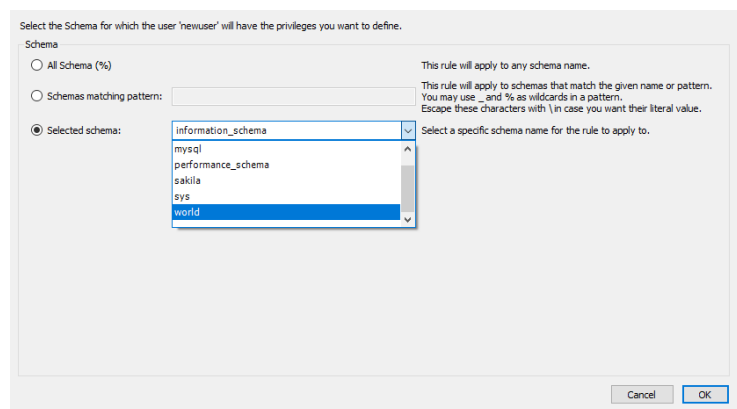
- a. Crear los siguientes usuarios:

<p>Usuario 1</p> <p>Nombre: user1.</p> <p>Contraseña:</p> <p>Permisos: SELECT</p>	<p>Usuario 2</p> <p>Nombre: user2.</p> <p>Permisos: SELECT, update, delete</p>	<p>Usuario 3</p> <p>Nombre: user3</p> <p>Permisos: SELECT, UPDATE, DELETE, LOCK TABLE</p>
---	--	---

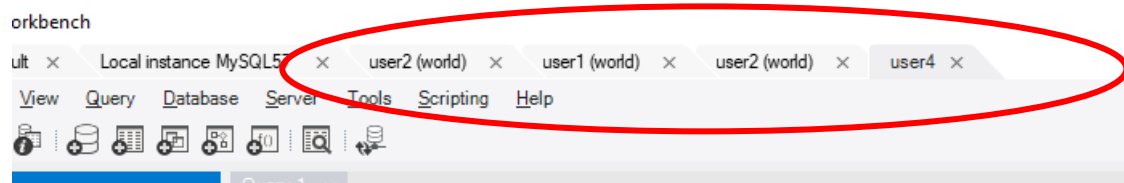
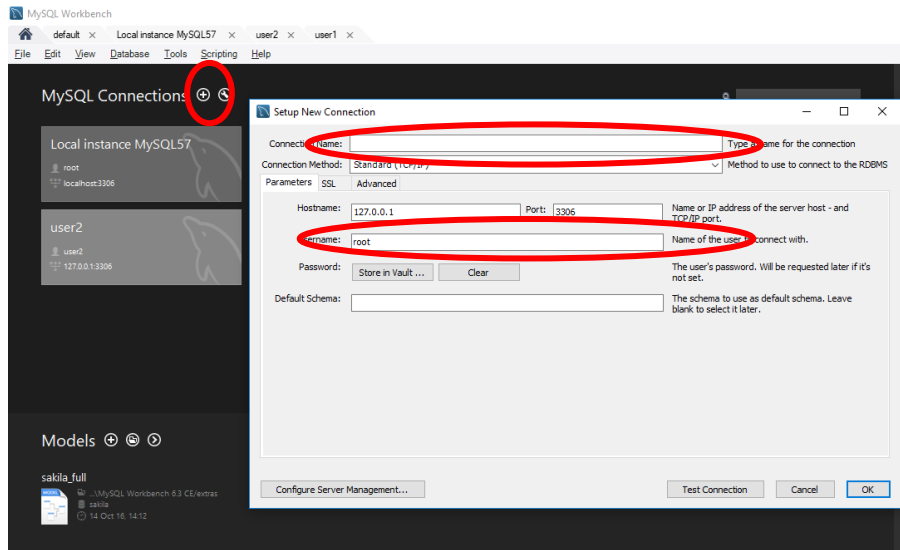
- b. Utilizar el usuario root en la aplicación Mysql Workbench, pulsar en “User and Privileges” y en “add account”:



- c. Pulsar en “schema privileges” y en “add entry”. Seleccionar el esquema “world”:



- d. Luego seleccionamos los privilegios que queremos dar a ese usuario y pulsamos en “Apply”.
- e. Abrir una pestaña con cada usuario utilizando el botón “home de Workbench”.

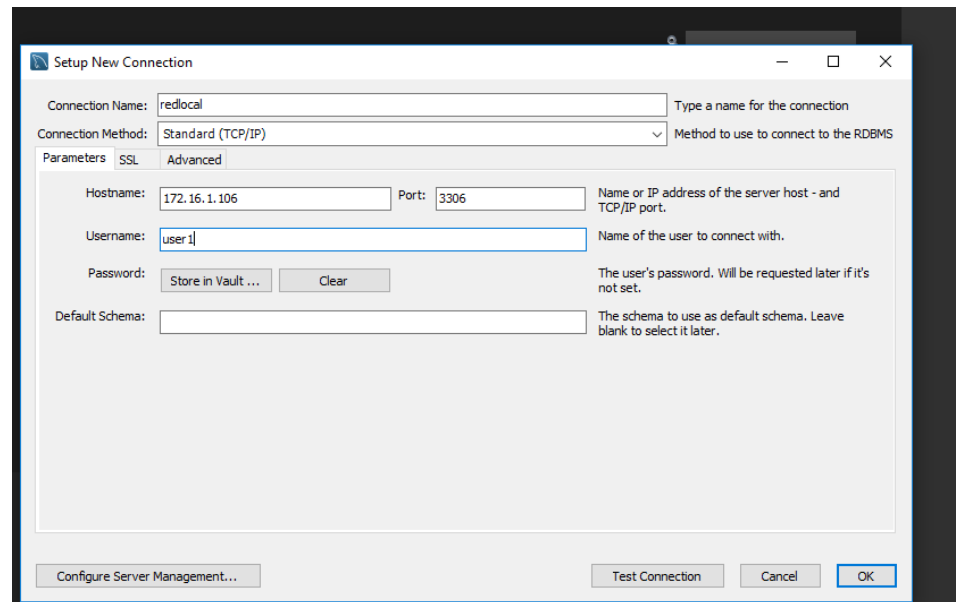


- f. Ejecutar para usuario1: “select * from city where countrycode='AFG';”
- g. Ejecutar para usuario2, con autocommit desactivado: “update city set District='kabolia1' where id=1;”
- h. Revisar la tabla con el usuario1: “select * from city where countrycode='AFG';”
- i. Ejecutar con usuario2: “commit;”
- j. Volver a revisar la tabla con usuario1
- k. Ahora realizaremos otro ejercicio diferente. Ejecuta con usuario2 y autocommit desactivado: “update city set District='kabolia2' where id=1;”
- l. Intentar borrar ese registro con el usuario3: “delete from city where id=1”. ¿Qué ha ocurrido?

10. Por último, vamos a estudiar el acceso concurrente a una base de datos. Es decir, que varios usuarios puedan acceder a una base de datos de una manera simultánea. Para ello, desde otro ordenador que tenga instalado Mysql Workbench y esté conectado en red local al nuestro, vamos a hacer la siguiente prueba:

- a. Crear una conexión con los siguientes datos:

1. Hostname: la dirección IP del servidor donde nos queramos conectar. La obtendremos haciendo “ipconfig” sobre la línea de comandos.



2. Usuario -> User1. Ejecutar: —> cmd en la línea de comando ipconfig.
3. Ahora desde dos ordenadores diferentes ejecutaremos las siguientes consultas:
 - i. Usuario 1 → ordenador 1 → select * from city;
 - ii. Usuario 2 → ordenador 2 →select * from city;
 - iii. update city set Name='amsterdam1' where id=5;
 - iv. Usuario 1 → select * from city;