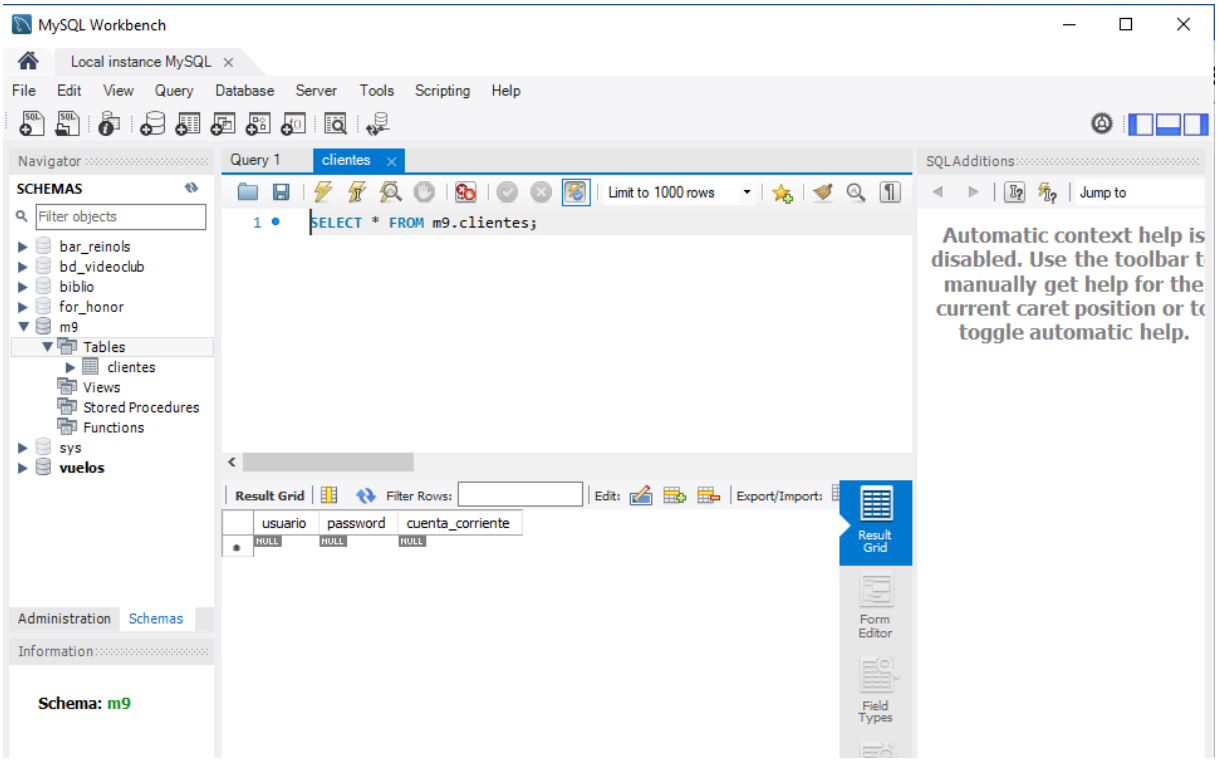
Práctica de cifrado con Bases de datos

Jonatan Valle Corrales – Sergio Bereño Verón

Vamos a poner en práctica los conocimientos que tenemos sobre encriptación de BBDD trabajando con Mysql. La práctica es sencilla, pero servirá de ejemplo.

1. Instala el XAMPP. Lo encontraras en \\marte\programari. Puede que necesites desactivar el UAC de Windows.

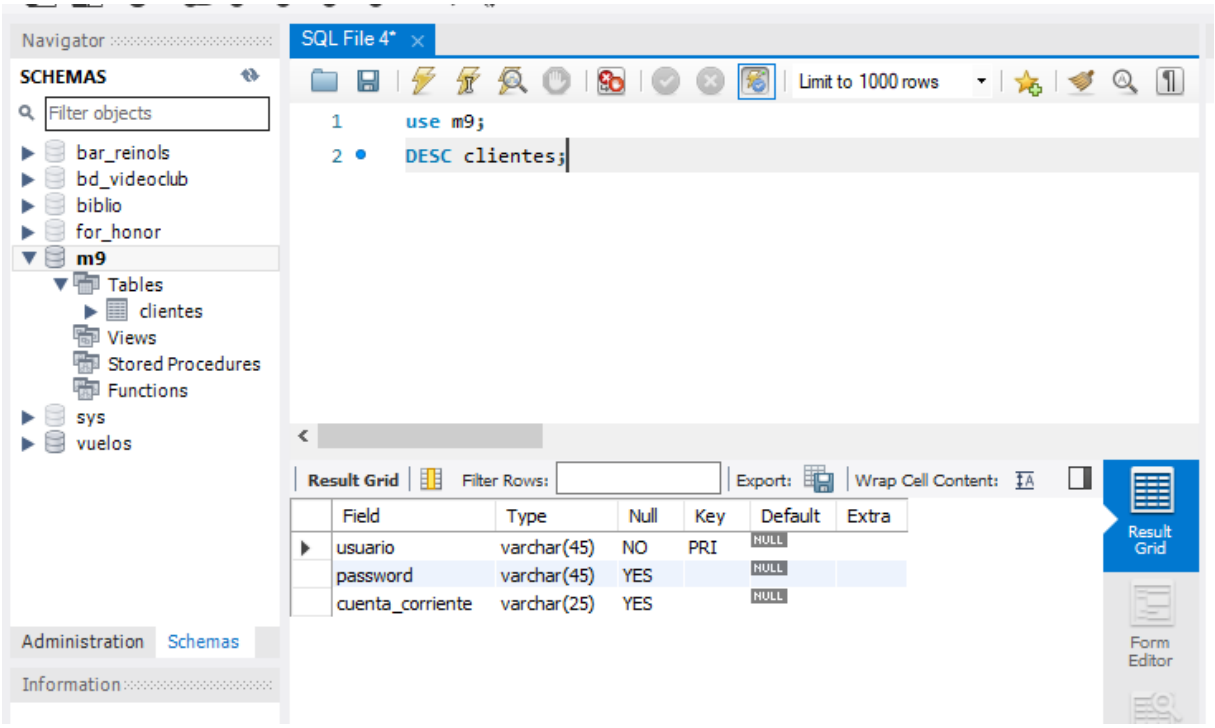
Utilizo MYSQL Workbench



2. Entra en el panel de control y comprueba que PHP y MySql están iniciados. Abre el navegador y entra en http://localhost/phpmyadmin o desde XAMPP control panel -> Admin.

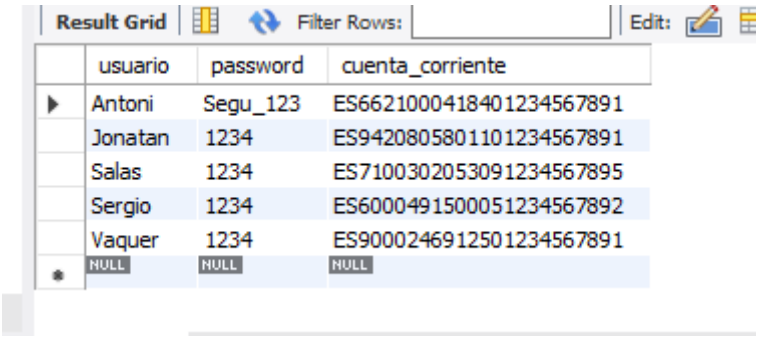
Utilizo MYSQL Workbench

3. Crea una tabla CLIENTES con tres campos: usuario, password y cuenta corriente. Pon los tipos, longitudes y restricciones adecuadas. Lista la estructura de la tabla.



4. Inserta 5 usuarios. Un usuario debe ser 'Antoni', password 'Segu_123'. Lista los 5 usuarios

```
INSERT INTO clientes (usuario, password, cuenta_corriente) VALUES ('Antoni', 1234, 'ES6621000418401234567891');
INSERT INTO clientes (usuario, password, cuenta_corriente) VALUES ('Sergio', 1234, 'ES6000491500051234567892');
INSERT INTO clientes (usuario, password, cuenta_corriente) VALUES ('Jonatan', 1234, 'ES9420805801101234567891');
INSERT INTO clientes (usuario, password, cuenta_corriente) VALUES ('Vaquer', 1234, 'ES9000246912501234567891');
INSERT INTO clientes (usuario, password, cuenta_corriente) VALUES ('Salas', 1234, 'ES7100302053091234567895');
```



5. Se quiere que el campo password esté protegido por un algoritmo de hash (MD5, SHA o SHA1). Indica el algoritmo escogido. Realiza las modificaciones pertinentes en la estructura de la base de datos. Indica las modificaciones.

   El algoritmo escogido es MD5.

   Hemos añadido una columna nueva para poder ver la diferencia entre un password sin hash a un password con hash.

```
ALTER TABLE clientes ADD password_encrypt VARCHAR(60);
```

   Hemos updateado la tabla para añadir el valor hash de la password a la nueva columna.

```
UPDATE clientes SET password_encrypt = md5(password) WHERE password_encrypt IS NULL;
```

6. Aplica el algoritmo de hash al campo password. Indica la sentencia para la transformación. Lista los 5 usuarios.

```
UPDATE clientes SET password_encrypt = md5(password) WHERE password_encrypt IS NULL;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

usuario	password	cuenta_corriente	password_encrypt
Antoni	Segu_123	ES6621000418401234567891	00098a5ef7f12c0b077ad6134fb52935
Jonatan	1234	ES9420805801101234567891	81dc9bdb52d04dc20036dbd8313ed055
Salas	1234	ES7100302053091234567895	81dc9bdb52d04dc20036dbd8313ed055
Sergio	1234	ES6000491500051234567892	81dc9bdb52d04dc20036dbd8313ed055
Vaquer	1234	ES9000246912501234567891	81dc9bdb52d04dc20036dbd8313ed055
NULL	NULL	NULL	NULL

7. Se quiere que el campo cuenta corriente esté encriptado. Realiza las modificaciones pertinentes en la estructura de la base de datos. Indica las modificaciones. Indica la encriptación que escoges.

   Hemos añadido una columna mas para poder ver la diferencia entre cuenta corriente sin encriptar y encriptada.

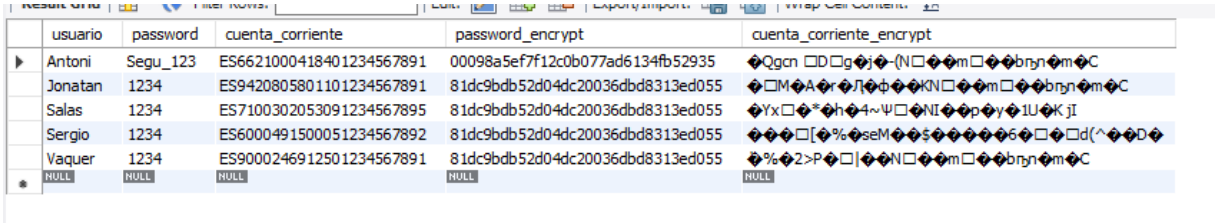
```
ALTER TABLE clientes ADD cuenta_corriente_encrypt VARBINARY(255);
```

La encriptación que hemos elegido es AES, y como contraseña m9.
Hemos updateado la tabla para añadir el valor encriptado a la nueva columna

```
UPDATE clientes SET cuenta_corriente_encrypt = AES_ENCRYPT(cuenta_corriente, 'm9') WHERE cuenta_corriente_encrypt IS NULL;
```

8. Aplica la encriptación al campo cuenta corriente. Indica la sentencia para la transformación. Lista los 5 usuarios.

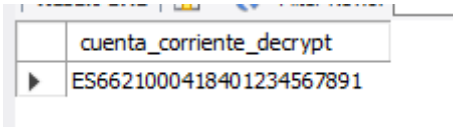
```
UPDATE clientes SET cuenta_corriente_encrypt = AES_ENCRYPT(cuenta_corriente, 'm9') WHERE cuenta_corriente_encrypt IS NULL;
```



usuario	password	cuenta_corriente	password_encrypt	cuenta_corriente_encrypt
Antoni	Segu_123	ES6621000418401234567891	00098a5ef7f12c0b077ad6134fb52935	◆Qgcn □D□g◆j◆-(N□◆◆m□◆◆bɲn◆m◆C
Jonatan	1234	ES9420805801101234567891	81dc9bdb52d04dc20036dbd8313ed055	◆□M◆A◆r◆Л◆ф◆◆KN□◆◆m□◆◆bɲn◆m◆C
Salas	1234	ES7100302053091234567895	81dc9bdb52d04dc20036dbd8313ed055	◆Yx□◆*◆h◆4~Ψ□◆NI◆◆p◆y◆1U◆K jI
Sergio	1234	ES6000491500051234567892	81dc9bdb52d04dc20036dbd8313ed055	◆◆◆□[◆%◆seM◆◆$◆◆◆◆◆6◆□◆□d(^◆◆D◆
Vaquer	1234	ES9000246912501234567891	81dc9bdb52d04dc20036dbd8313ed055	◆%◆2>P◆□|◆◆N□◆◆m□◆◆bɲn◆m◆C
NULL	NULL	NULL	NULL	NULL

9. Deseamos que a partir del usuario 'Antoni' y password 'Segu_123' obtengamos la cuenta corriente, desencriptada por supuesto, de ese usuario. Indica la sentencia utilizada y muestra el resultado.

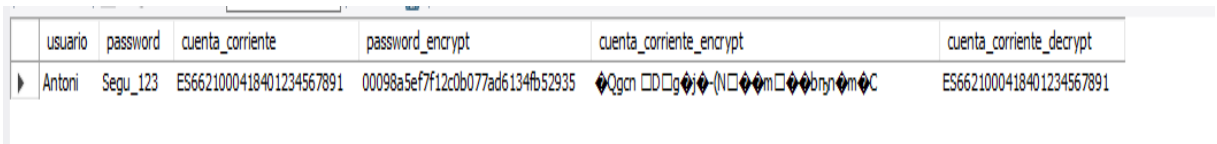
```
SELECT AES_DECRYPT(cuenta_corriente_encrypt, 'm9') as cuenta_corriente_decrypt FROM clientes WHERE usuario LIKE 'Antoni';
```



cuenta_corriente_decrypt
ES6621000418401234567891

Lo hemos puesto de esta manera para que lo puedas visualizar mejor.

```
SELECT *, AES_DECRYPT(cuenta_corriente_encrypt, 'm9') as cuenta_corriente_decrypt FROM clientes WHERE usuario LIKE 'Antoni' AND password_encrypt = MD5('Segu_123');
```



usuario	password	cuenta_corriente	password_encrypt	cuenta_corriente_encrypt	cuenta_corriente_decrypt
Antoni	Segu_123	ES6621000418401234567891	00098a5ef7f12c0b077ad6134fb52935	◆Qgcn □D□g◆j◆-(N□◆◆m□◆◆bɲn◆m◆C	ES6621000418401234567891