

```

1 • CREATE USER userusragain IDENTIFIED BY '123456';
2 • GRANT ALL PRIVILEGES ON * . * TO 'userusragain';
3 • CREATE SCHEMA `userusragainchema` ;

```

Output

Action Output

#	Time	Action	Message
✓ 1	17:54:38	CREATE USER userusragain IDENTIFIED BY '123456'	0 row(s) affected
✓ 2	17:54:38	GRANT ALL PRIVILEGES ON * . * TO 'userusragain'	0 row(s) affected
✓ 3	17:54:38	CREATE SCHEMA `userusragainchema`	1 row(s) affected

Setup New Connection

Connection Name: UserRegistroConn Type a name for the connection

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: 127.0.0.1 Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: userusragain Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.

Default Schema: userusragainchema The schema to use as default schema. Leave blank to select it later.

MySQL Workbench

i Successfully made the MySQL connection

Information related to this connection:

Host: 127.0.0.1
Port: 3306
User: userusragain
SSL: enabled with TLS_AES_128_GCM_SHA256

A successful MySQL connection was made with the parameters defined for this connection.

Test Connection Cancel OK

OK

```

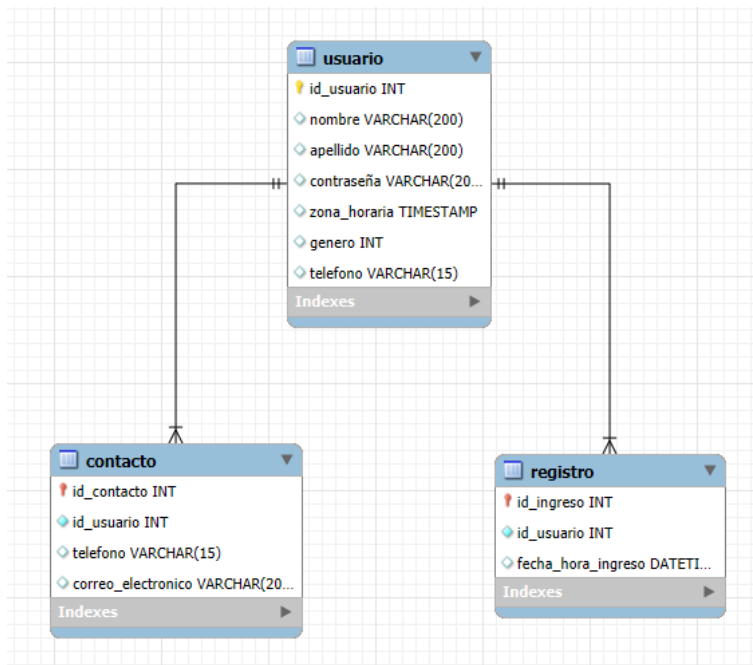
1 • CREATE TABLE usuario(
2     id_usuario INT UNSIGNED NOT NULL PRIMARY KEY,
3     nombre VARCHAR(200),
4     apellido VARCHAR(200),
5     contraseña VARCHAR(200),
6     zona_horaria TIMESTAMP,
7     genero INTEGER,
8     telefono VARCHAR(15)
9 );
10 • CREATE TABLE registro(
11     id_ingreso INT UNSIGNED NOT NULL PRIMARY KEY,
12     id_usuario INT UNSIGNED NOT NULL,
13     fecha_hora_ingreso DATETIME
14 );
15 • CREATE TABLE contacto(
16     id_contacto INT UNSIGNED NOT NULL PRIMARY KEY,
17     id_usuario INT UNSIGNED NOT NULL,
18     telefono VARCHAR(15),
19     correo_electronico VARCHAR(200)
20 );

```

Output

Action Output

#	Time	Action
✓ 1	18:39:36	CREATE TABLE usuario(id_usuario INT UNSIGNED NOT NULL PRII
✓ 2	18:39:36	CREATE TABLE registro(id_ingreso INT UNSIGNED NOT NULL PRII
✓ 3	18:39:36	CREATE TABLE contacto(id_contacto INT UNSIGNED NOT NULL F



```

21  /* Si un usuario tiene muchos registros posibles, y un usuario tiene uno o más contactos
22  * se tiene las siguientes alteraciones en las tablas cuestión de relacionarlas según el
23  * supuesto de partida
24  */
25  • ALTER TABLE registro ADD CONSTRAINT fk_registro_usuario FOREIGN KEY (id_ingreso) REFERENCES usuario (id_usuario);
26  • ALTER TABLE contacto ADD CONSTRAINT fk_contacto_usuario FOREIGN KEY (id_contacto) REFERENCES usuario (id_usuario);
  
```

```

27  • SELECT NOW();
28  • SELECT CURRENT_TIMESTAMP; /* SELECT CURRENT_TIMESTAMP();*/
29  • SELECT TIMEDIFF(NOW(), UTC_TIMESTAMP); /*-04:00:00*/
30  • SET time_zone = '-02:00';
31  • SELECT TIMEDIFF(NOW(), UTC_TIMESTAMP);/*-02:00:00*/
  
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
TIMEDIFF(NOW(), UTC_TIMESTAMP)			
-02:00:00			

```

32  • SELECT * FROM usuario;
33  • INSERT INTO usuario VALUES (1, 'Usuario 001', 'Usuario 001', '12345678', CURRENT_TIMESTAMP, 1, '+56987654321');
34  • INSERT INTO usuario VALUES (2, 'Usuario 002', 'Usuario 002', '12345678', CURRENT_TIMESTAMP, 1, '+56987654321');
35  • INSERT INTO usuario VALUES (3, 'Usuario 003', 'Usuario 003', '12345678', CURRENT_TIMESTAMP, 2, '+56987654321');
36  • INSERT INTO usuario VALUES (4, 'Usuario 004', 'Usuario 004', '12345678', CURRENT_TIMESTAMP, 2, '+56987654321');
37  • INSERT INTO usuario VALUES (5, 'Usuario 005', 'Usuario 005', '12345678', CURRENT_TIMESTAMP, 1, '+56987654321');
38  • INSERT INTO usuario VALUES (6, 'Usuario 006', 'Usuario 006', '12345678', CURRENT_TIMESTAMP, 1, '+56987654321');
39  • INSERT INTO usuario VALUES (7, 'Usuario 007', 'Usuario 007', '12345678', CURRENT_TIMESTAMP, 2, '+56987654321');
40  • INSERT INTO usuario VALUES (8, 'Usuario 008', 'Usuario 008', '12345678', CURRENT_TIMESTAMP, 1, '+56987654321');
  
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	id_usuario	nombre	apellido	contraseña	zona_horaria	genero	telefono
▶	1	Usuario 001	Usuario 001	12345678	2024-08-16 21:05:39	1	+56987654321
	2	Usuario 002	Usuario 002	12345678	2024-08-16 21:08:11	1	+56987654321
	3	Usuario 003	Usuario 003	12345678	2024-08-16 21:08:11	2	+56987654321
	4	Usuario 004	Usuario 004	12345678	2024-08-16 21:08:11	2	+56987654321
	5	Usuario 005	Usuario 005	12345678	2024-08-16 21:08:11	1	+56987654321
	6	Usuario 006	Usuario 006	12345678	2024-08-16 21:08:11	1	+56987654321
	7	Usuario 007	Usuario 007	12345678	2024-08-16 21:08:11	2	+56987654321
	8	Usuario 008	Usuario 008	12345678	2024-08-16 21:08:11	1	+56987654321
⬅	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```

41 • SELECT * FROM registro;
42 • INSERT INTO registro VALUES (1, 1, NOW());
43 • INSERT INTO registro VALUES (2, 2, NOW());
44 • INSERT INTO registro VALUES (3, 3, NOW());
45 • INSERT INTO registro VALUES (4, 4, NOW());
46 • INSERT INTO registro VALUES (5, 5, NOW());
47 • INSERT INTO registro VALUES (6, 6, NOW());
48 • INSERT INTO registro VALUES (7, 7, NOW());
49 • INSERT INTO registro VALUES (8, 8, NOW());
50

```

Result Grid			
Filter Rows:			
	id_ingreso	id_usuario	fecha_hora_ingreso
▶	1	1	2024-08-16 21:14:50
	2	2	2024-08-16 21:15:26
	3	3	2024-08-16 21:15:26
	4	4	2024-08-16 21:15:26
	5	5	2024-08-16 21:15:26
	6	6	2024-08-16 21:15:26
	7	7	2024-08-16 21:15:26
	8	8	2024-08-16 21:15:26
✱	NULL	NULL	NULL

```

50 • SELECT * FROM contacto;
51 • INSERT INTO contacto VALUES (1, 1, '+56987654321', 'correo@correo.com');
52 • INSERT INTO contacto VALUES (2, 2, '+56987654321', 'correo@correo.com');
53 • INSERT INTO contacto VALUES (3, 3, '+56987654321', 'correo@correo.com');
54 • INSERT INTO contacto VALUES (4, 4, '+56987654321', 'correo@correo.com');
55 • INSERT INTO contacto VALUES (5, 5, '+56987654321', 'correo@correo.com');
56 • INSERT INTO contacto VALUES (6, 6, '+56987654321', 'correo@correo.com');
57 • INSERT INTO contacto VALUES (7, 7, '+56987654321', 'correo@correo.com');
58 • INSERT INTO contacto VALUES (8, 8, '+56987654321', 'correo@correo.com');

```

Result Grid				
Filter Rows:				
	id_contacto	id_usuario	telefono	correo_electronico
▶	1	1	+56987654321	correo@correo.com
	2	2	+56987654321	correo@correo.com
	3	3	+56987654321	correo@correo.com
	4	4	+56987654321	correo@correo.com
	5	5	+56987654321	correo@correo.com
	6	6	+56987654321	correo@correo.com
	7	7	+56987654321	correo@correo.com
	8	8	+56987654321	correo@correo.com
✱	NULL	NULL	NULL	NULL

Sí, principalmente por que al almacenar la zona horaria como se ha realizado es útil si lo que quieres es tener un mayor control sobre cómo se manejan las conversiones y presentaciones de las fechas y horas. Usar **TIMESTAMP** con Zona Horaria integrada puede simplificar el manejo de ellas, ya que MySQL se encargará automáticamente de las conversiones al almacenar y recuperar los valores. Es importante asegurarse de que la configuración de zonas horarias en el servidor esté correctamente ajustada, cuestión que está por cierto solicitada en el ejercicio (a modo de). Cosa semejante ocurre con **DATETIME**, en este caso como la tabla que contiene esa columna con ese tipo, no es exactamente el mismo contexto, de cuando un usuario se registra, está bien aplicado, si bien es más costoso (**DATETIME** ocupa el doble que **TIMESTAMP** y podría ser más lento), es específicamente mejor aplicado al modelo y a lo que se entiende debe representar y hacer cada tabla (aunque este acercamiento es sensible y técnicamente correcto, podrían cambiar las condiciones que provoquen que el modelo deba ser refactorizado, pero por ahora está bien).

