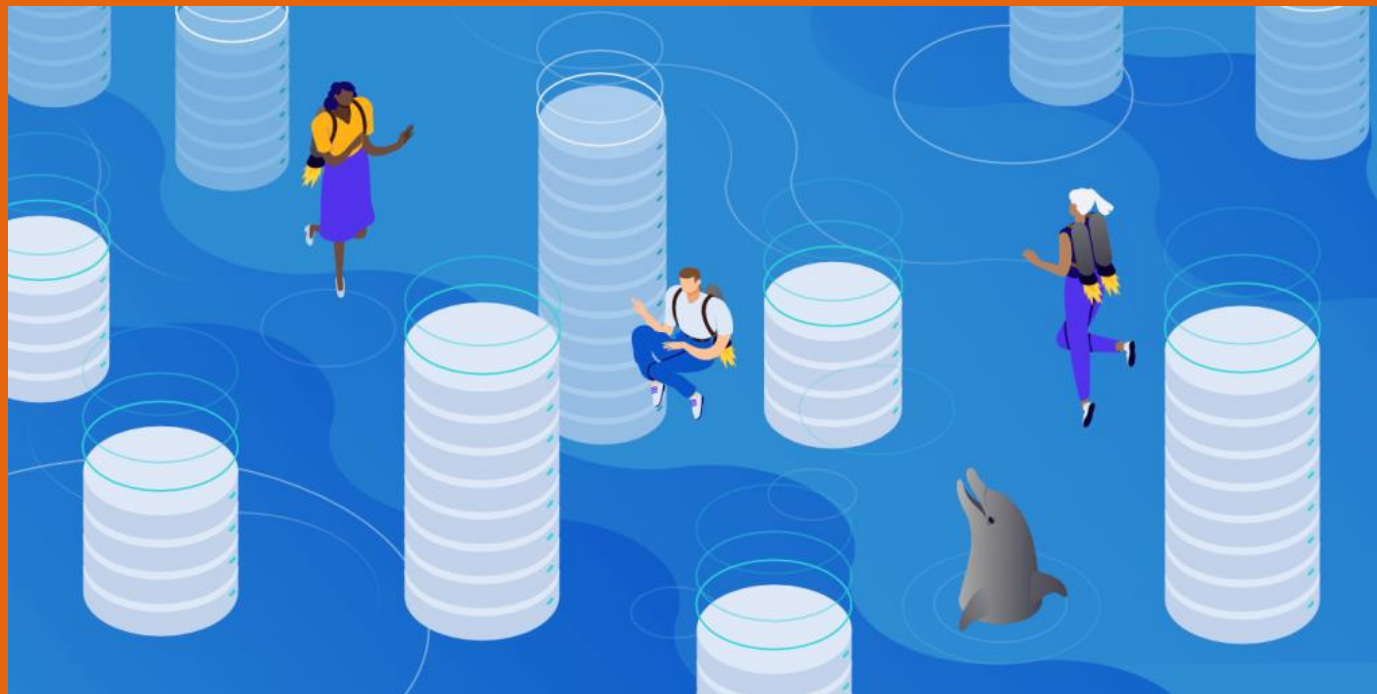


METODOLOGÍA DE LA INVESTIGACIÓN

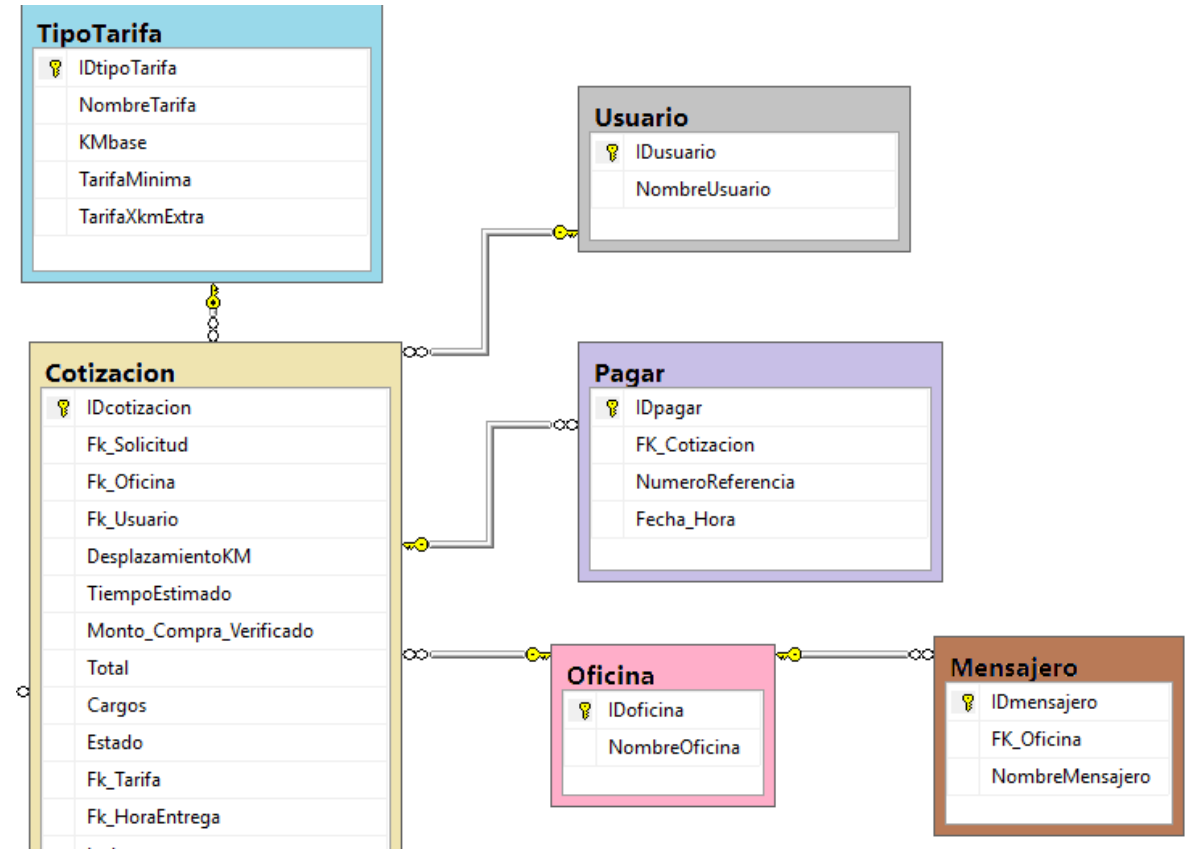
Clase 11

Prof. Natalia Lucero

MYSQL

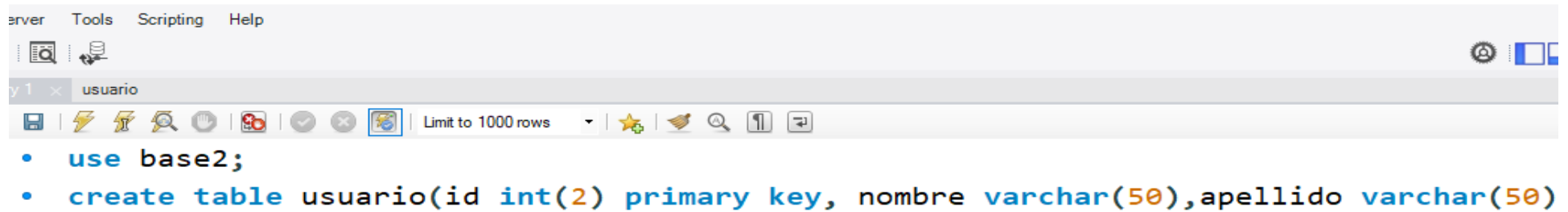


TABLAS



USE

- Nos permite posicionarnos en una base de datos
- Use <nombre de la base>;



CARGAR DATOS EN LAS TABLAS

- Para cargar datos a nuestras tablas utilizaremos también una sentencia SQL propiamente del lenguaje a partir de palabras reservadas, en este caso utilizaremos:

Insert Into <nombre de la tabla> values (<valores>);

- Para tener en cuenta los datos en números se escriben en números
- Los datos tipo caracter o varchar se escriben entre 1 comilla 'Nombre'

UNA VEZ QUE UTILIZAMOS LOS COMANDOS INSERT INTO CON TODOS LOS VALORES DE NUESTRA TABLA, EJECUTAREMOS



MySQL Workbench

Metodología x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

base1

Tables

usuario

Columns

Indexes

Foreign Keys

Triggers

Views

Administration Schemas

Information

Table: usuario

Columns:

id int PK

nombre varchar(50)

apellido varchar(50)

correo varchar(100)

Query 1 usuario

```
1 use base1;
2 insert into usuario values (1, 'Walter', 'Rojas', 'WalterRojas@gmail.com');
3
```

Limit to 1000 rows

Output

Action Output

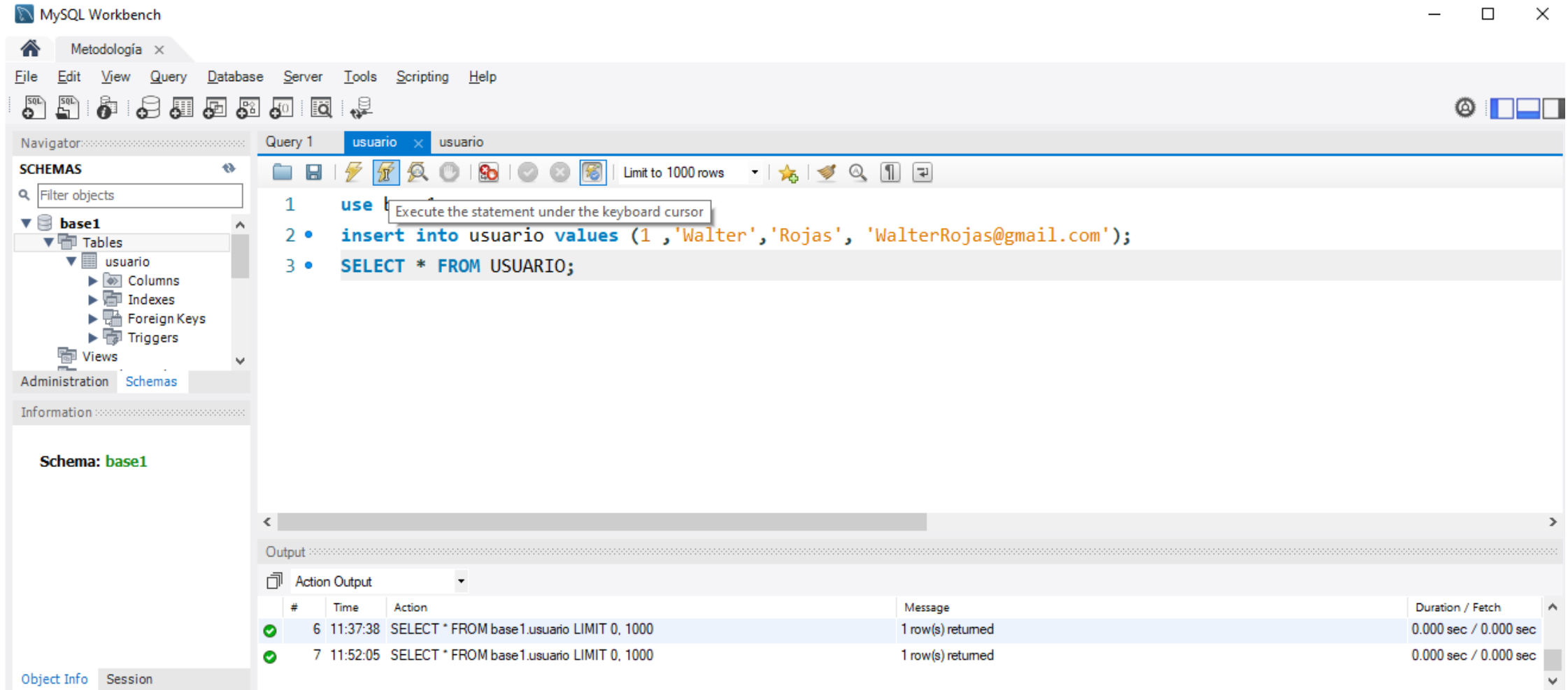
#	Time	Action	Message	Duration / Fetch
✓ 4	11:37:14	use base1	0 row(s) affected	0.000 sec
✓ 5	11:37:14	insert into usuario values (1, 'Walter', 'Rojas', 'WalterRojas@gmail.com')	1 row(s) affected	0.015 sec

Object Info Session

SENTANCIA SELECT

- UTILIZAREMOS LA SENTENCIA SELECT *FROM PARA VISUALIZAR LOS DATOS INGRESADOS EN NUESTRA TABLA, EN ESTE CASO AL EJECUTAR LA SENTENCIA UTILIZAREMOS EL RAYO CON EL CURSOR , PARA EJECUTAR SÓLO LA ULTIMA LÍNEA
- SELECT *FROM <nombre de la tabla>;

EJECUTANDO LA SENTENCIA SELECT * FROM PERO CON EL RAYO QUE TIENE EL CURSO , PORQUE SÓLO QUEREMOS EJECUTAR LA ÚLTIMA LÍNEA DE COMANDOS.



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view showing 'base1' and its tables, including 'usuario'. The main editor window shows a SQL query with three lines:

```
1 use   
2 • insert into usuario values (1 , 'Walter', 'Rojas', 'WalterRojas@gmail.com');  
3 • SELECT * FROM USUARIO;
```

A tooltip 'Execute the statement under the keyboard cursor' is visible over the 'use' command. The 'Output' panel at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
✓ 6	11:37:38	SELECT * FROM base1.usuario LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
✓ 7	11:52:05	SELECT * FROM base1.usuario LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

UNA VEZ EJECUTADO EL COMANDO, NO APARECERÁ LA TABLA CON LOS DATOS INGRESADOS

The screenshot shows the MySQL Workbench interface. On the left, the 'Navigator' pane shows the 'base1' schema with a 'usuario' table. The 'Query 1' editor contains the following SQL commands:

```
1 use base1;
2 • insert into usuario values (1 , 'Walter', 'Rojas', 'WalterRojas@gmail.com');
3 • SELECT * FROM USUARIO;
4
```

The 'Result Grid' shows the results of the query:

	id	nombre	apellido	correo
▶	1	Walter	Rojas	WalterRojas@gmail.com
*	NULL	NULL	NULL	NULL

The 'Output' pane at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
✓ 7	11:52:05	SELECT * FROM base1.usuario LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
✓ 8	11:53:10	SELECT * FROM USUARIO LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

ACTIVIDAD 1

- Generamos datos para nuestras tablas creadas en cada base de dato de la clase nro.10, añadiendo tres filas de datos a cada una de ellas.
- Utilizar sentencias:
- Use
- Inser Into
- Select * From

AÑADIENDO MÁS DATOS A LAS TABLAS CREADAS DENTRO DE LAS BASES DE DATOS DE LA CLASE 10

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'base1' selected, containing a table named 'usuario'. The main editor window shows a SQL query with three lines: 'use base1;', 'insert into usuario values (2, 'Romina', 'Tello', 'RominaT@gmail.com');', and 'SELECT * FROM USUARIO;'. Below the query editor, the 'Result Grid' shows the data from the 'usuario' table. The table has four columns: 'id', 'nombre', 'apellido', and 'correo'. It contains two rows: one for 'Walter Rojas' and one for 'Romina Tello'. The bottom panel shows the 'Output' window with 'Action Output' selected, displaying the execution log for the insert and select queries.

Query 1 usuario x usuario

```
1 use base1;
2 insert into usuario values (2, 'Romina', 'Tello', 'RominaT@gmail.com');
3 SELECT * FROM USUARIO;
```

Result Grid

	id	nombre	apellido	correo
▶	1	Walter	Rojas	WalterRojas@gmail.com
	2	Romina	Tello	RominaT@gmail.com
*	NULL	NULL	NULL	NULL

USUARIO 3 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 10	12:14:43	insert into usuario values (2, 'Romina', 'Tello', 'RominaT@gmail.com')	1 row(s) affected	0.000 sec
✓ 11	12:14:48	SELECT * FROM USUARIO LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

SENTENCIA SELECT

- La sentencia SELECT es para mostrar, para ver, para visualizar datos de una base de datos.



SELECT <COLUMNA> FROM <NOMBRE BASE.NOMBRE TABLA>;

Nos permite traer los datos específicos cuando no utilizamos el asterisco y seleccionando sólo los datos que necesitamos

The screenshot shows the MySQL Workbench interface. The left sidebar contains the 'Navigator' panel with 'SCHEMAS' and 'base1' expanded, showing 'usuario' table. The main editor shows a SQL query: `use base1;` and `SELECT NOMBRE FROM base1.usuario;`. The 'Result Grid' shows two rows: 'Walter' and 'Romina'. The 'Output' panel at the bottom shows the execution log.

Query 1 usuario x usuario

```
1 use base1;
2 • SELECT NOMBRE FROM base1.usuario;
3
4
```

Result Grid

	NOMBRE
▶	Walter
	Romina

Output

usuario 4 x Read Only

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 12	12:39:39	use base1	0 row(s) affected	0.000 sec
✓ 13	12:39:39	SELECT NOMBRE FROM base1.usuario LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Query Completed

NOS PERMITE TRAER DATOS DE DIFERENTES COLUMNAS

SELECT <COMULMNA1, COLUMNA 3, COLUMNA 5 FROM <NOMBRE DE LA BASE>.<NMBRE DE LA TABLA>;

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left displays the database structure, including the 'base1' database and the 'usuario' table. The 'Query' pane in the center contains the following SQL code:

```
1 use base1;
2 • SELECT id, nombre, correo FROM base1.usuario;
3
4
```

The 'Result Grid' pane shows the results of the query, which returned 2 rows:

	id	nombre	correo
1	1	Walter	WalterRojas@gmail.com
2	2	Romina	RominaT@gmail.com

The 'Table: usuario' section in the 'Information' pane lists the columns: id (int PK), nombre (varchar(50)), apellido (varchar(50)), and correo (varchar(100)). The 'Output' pane at the bottom shows the execution log, indicating that the query was successful and returned 2 rows.

Query Completed

SELECT TAMBIÉN NOS PERMITE CAMBIAR EL NOMBRE A UNA COLUMNA

UTILIZAREMOS

Select <Nombre de la columna a> AS <nuevo nombre de la columna> FROM < Nombre dela Base> . <Nombre de tabla>;

MySQL Workbench interface showing a query execution. The query is:

```
1 use base1;
2 SELECT nombre as Nombre from base1.usuario;
```

The result grid displays the following data:

Nombre
Walter
Romina

The output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
15	13:04:42	use base1	0 row(s) affected	0.000 sec
16	13:04:42	SELECT nombre as Nombre from base1.usuario LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

PARA CAMBIAR EL NOMBRE A MÚLTIPLES COLUMNAS

Select <Nombre de la columna A > AS <nuevo nombre de la columna A>, <nombre de columna B> AS < nuevo nombre columna B > FROM < Nombre dela Base> . <Nombre de tabla>;

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a database named 'base1' with a table 'usuario'. The 'Table: usuario' section lists columns: 'id' (int PK), 'nombre' (varchar(50)), 'apellido' (varchar(50)), and 'correo' (varchar(100)).

The main query editor shows the following SQL query:

```
1 use base1;  
2 SELECT nombre as Nombre , id as ID, apellido as Apellido, correo as Correo from base1.usuario;
```

Below the query editor, the 'Result Grid' shows the results of the query:

	Nombre	ID	Apellido	Correo
▶	Walter	1	Rojas	WalterRojas@gmail.com
	Romina	2	Tello	RominaT@gmail.com

At the bottom, the 'Action Output' pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
✓ 17	13:12:37	use base1	0 row(s) affected	0.000 sec
✓ 18	13:12:37	SELECT nombre as Nombre , id as ID, apellido as Apellido, correo as Correo from base...	2 row(s) returned	0.000 sec / 0.000 sec

SENTENCIA WHERE

Esta sentencia me traer datos específicos según lo que se le especifique en la búsqueda

SELECT * FROM base1.usuario where nombre = "Romina";

The screenshot displays the MySQL Workbench interface. On the left, the 'Navigator' pane shows a tree view of databases (base1, base2, base3) and their schemas. The 'base1' database is expanded, showing tables, views, stored procedures, and functions. The 'usuario' table is selected. Below the navigator, the 'Information' pane shows the table's structure: 'id' (int PK), 'nombre' (varchar(50)), 'apellido' (varchar(50)), and 'correo' (varchar(100)).

The main workspace shows a query editor with the following SQL statement:

```
SELECT * FROM base1.usuario where nombre = "Romina";
```

Below the query editor, the 'Result Grid' pane displays the results of the query. The results are shown in a table with 4 columns: id, nombre, apellido, and correo. The first row is highlighted, showing the user 'Romina Tello' with email 'RominaT@gmail.com'.

	id	nombre	apellido	correo
1	2	Romina	Tello	RominaT@gmail.com

At the bottom, the 'Output' pane shows the 'Action Output' table, which records the execution of the query. It includes columns for #, Time, Action, Message, and Duration / Fetch. The first row shows the execution of the query, returning 2 rows in 0.000 seconds.

#	Time	Action	Message	Duration / Fetch
1	14:45:01	SELECT * FROM base1.usuario LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

SENTENCIA DISTINCT

Esta sentencia me va a permitir mostrar datos distintos de una sola columna, obviará los datos repetidos

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a database named 'base1' with a table named 'usuario'. The 'Information' pane shows the structure of the 'usuario' table:

Column	Type
id	int PK
nombre	varchar(50)
apellido	varchar(50)
correo	varchar(100)

The 'Query' editor shows the following SQL statement:

```
1 • SELECT distinct (Apellido) from base1.usuario;
```

The 'Result Grid' shows the output of the query:

Apellido
Rojas
Tello

The 'Output' pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
3	14:55:15	SELECT * FROM base1.usuario LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
4	14:59:34	SELECT distinct (Apellido) from base1.usuario LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

At the bottom right, there is a watermark that says 'Activar Windows'.

ACTIVIDAD

- 1- Responder el cuestionario para la asistencia en el campus
- 2- Realizar la Actividad nº 1
- 3- Ejercitar las sentencias vista durante la clase