

TECNO SOLUTIONS



PABLO OCAÑO
IGNACIO BRESSANINI
FERNANDO PAJÓN
M. SOLEDAD CAUDANA

13-06-2024

DER

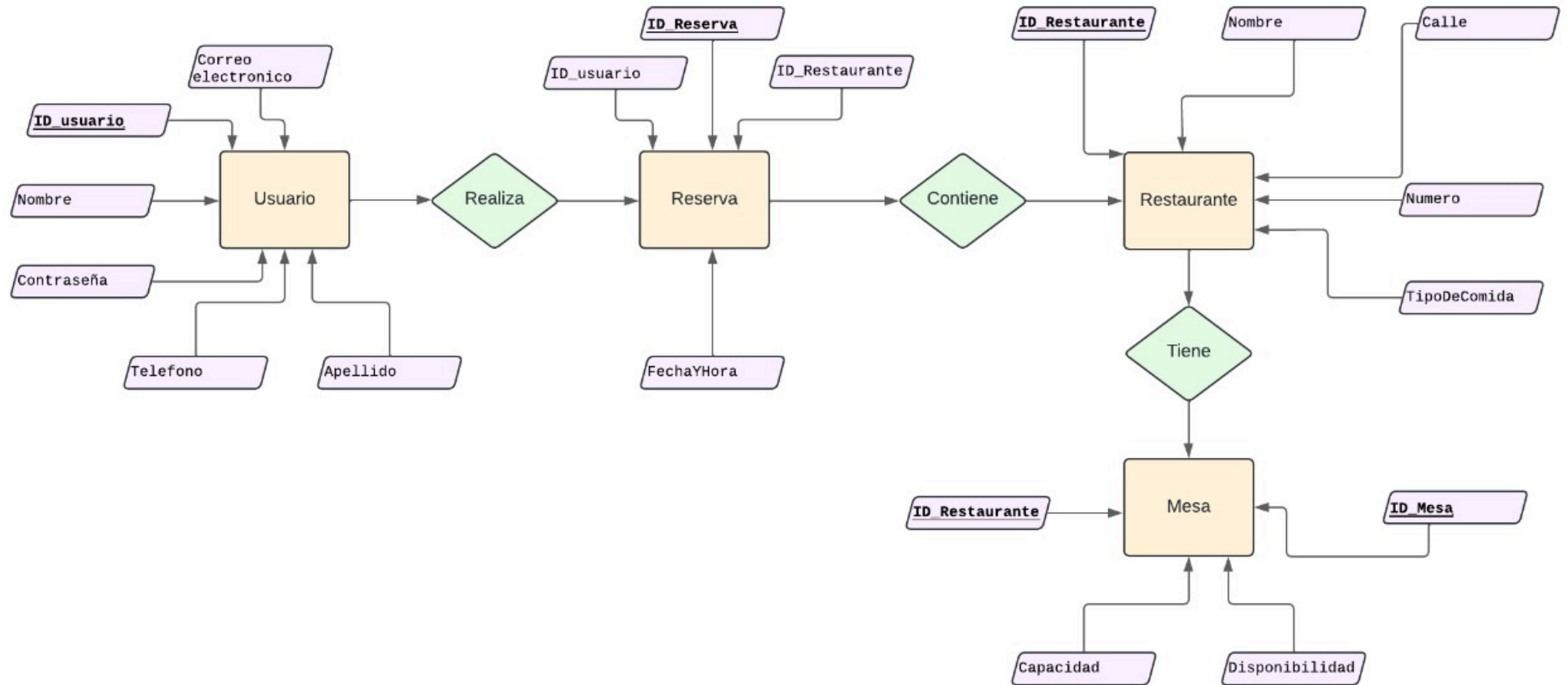


DIAGRAMA CROW FOOT

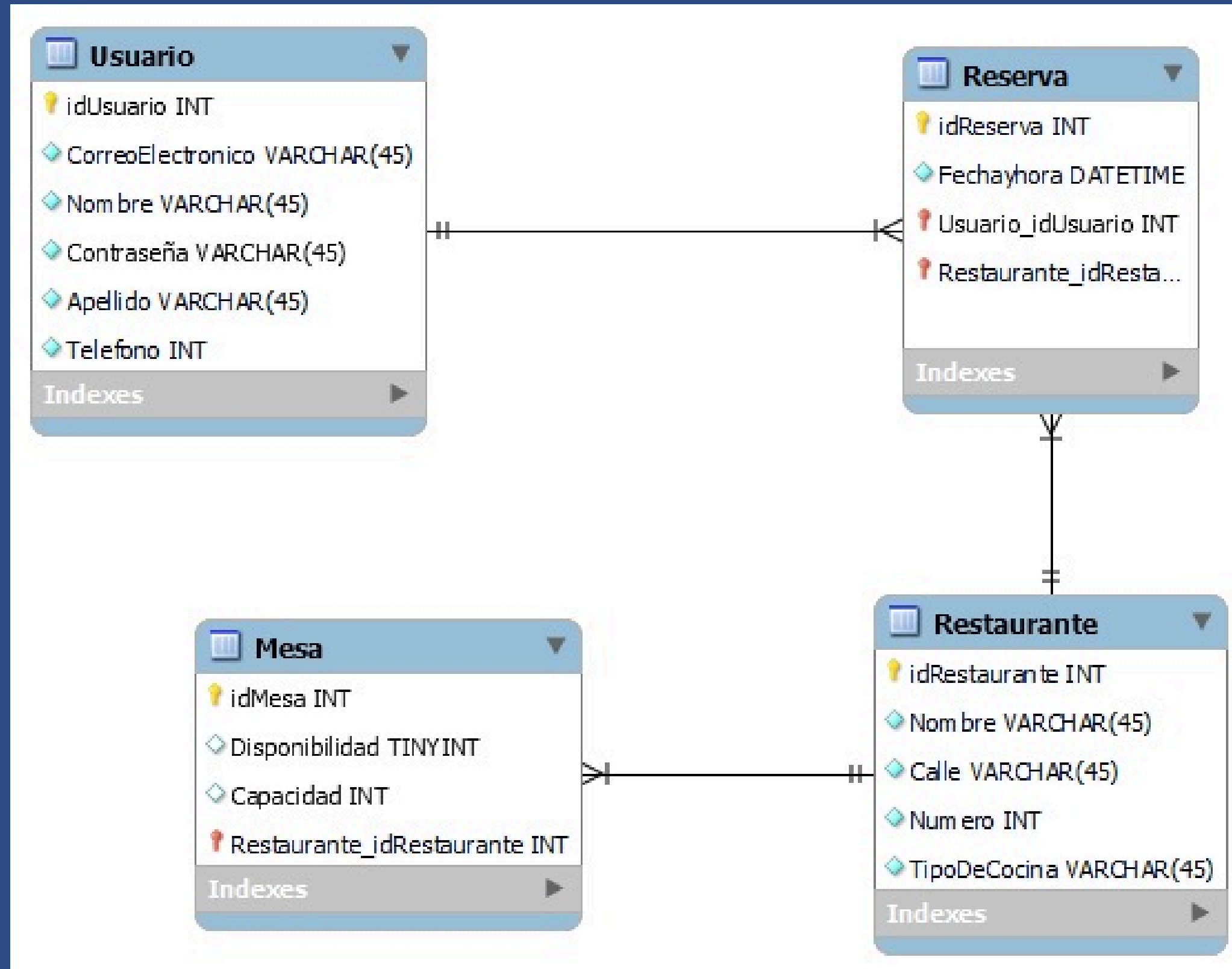


TABLA USUARIO

	usuario_id	email	contraseña	Nombre	Apellido	telefono
▶	1	pabblloocanno@gmail.com	pablo1	Pablo	Ocaño	2147483647
	2	soledad.caudana@gmail.com	sole1	Soledad	Caudana	2147483647
	3	fernandopajon06@gmail.com	fer1	Fernando	Pajon	2147483647
	4	nachobressanini1@gmail.com	nacho1	Ignacio	Bressanini	2147483647

TABLA RESERVA

	reserva_id	fecha	hora	usuario_id	restaurante_id
▶	1	2024-06-13	08:00:00	1	2
	2	2024-06-13	09:00:00	2	3
	3	2024-06-13	10:00:00	3	4
	4	2024-06-13	11:00:00	4	5
	5	2024-06-13	12:00:00	5	2
	6	2024-06-13	13:00:00	6	4

TABLA RESTAURANTE

	restaurante_id	nombre	calle	numero	tipo_de_comida
►	1	Bohemia Mexican	Av Pueyrredon	1345	Mexicana
	2	Bohemia Chinese	Gral Paz	72	China
	3	Bohemia Arabian	Av 24 de Septiembre	791	Arabe
	4	Bohemia Burguer	Obispo Trejo	1200	Hamburguesa
	5	Bohemia Italian	Av Colon	2840	Italiana
	6	Bohemia Indian	Rondeu	439	India
	7	Bohemia Grill	Rafael Nuñez	3248	Parrilla
	8	Bohemia Fish & Shellfish	Sucre	2504	Pescado y Marisco

TABLA MESA

	mesa_id	capacidad	disponibilidad	restaurante_id
▶	1	2	1	1
	2	4	1	1
	3	5	1	1
	4	6	1	1
	5	2	1	2
	6	4	1	2

INSERTAR DATOS

INSERT:

Sentencia SQL INSERT INTO que se utiliza para insertar nuevos registros en una tabla de mesa con los campos capacidad, disponibilidad y restaurante_id

```
INSERT INTO mesa (capacidad, disponibilidad, restaurante_id)
VALUES (3, 1, 9), (6, 1, 9), (8, 1, 9), (10, 1, 9),
       (3, 1, 10), (6, 1, 10), (8, 1, 10), (10, 1, 10),
       (3, 1, 11), (6, 1, 11), (8, 1, 11), (10, 1, 11),
       (3, 1, 12), (6, 1, 12), (8, 1, 12), (10, 1, 12),
       (3, 1, 13), (6, 1, 13), (8, 1, 13), (10, 1, 13),
       (3, 1, 14), (6, 1, 14), (8, 1, 14), (10, 1, 14),
       (3, 1, 15), (6, 1, 15), (8, 1, 15), (10, 1, 15),
       (3, 1, 16), (6, 1, 16), (8, 1, 16), (10, 1, 16);
```

The screenshot shows a database management interface with a Navigator pane on the left and a Query Editor on the right. The Navigator pane displays a tree structure of schemas, including 'reserva_rest' and 'reserva_test'. The 'reserva_test' schema is expanded, showing a table named 'mesa' with columns 'mesa_id', 'capacidad', 'disponibilidad', and 'restaurante_id'. The Query Editor shows the execution of the SQL statement: `INSERT INTO mesa (capacidad, disponibilidad, restaurante_id) VALUES (3, 1, 9), (6, 1, 9), (8, 1, 9), (10, 1, 9), (3, 1, 10), (6, 1, 10), (8, 1, 10), (10, 1, 10), (3, 1, 11), (6, 1, 11), (8, 1, 11), (10, 1, 11), (3, 1, 12), (6, 1, 12), (8, 1, 12), (10, 1, 12), (3, 1, 13), (6, 1, 13), (8, 1, 13), (10, 1, 13), (3, 1, 14), (6, 1, 14), (8, 1, 14), (10, 1, 14), (3, 1, 15), (6, 1, 15), (8, 1, 15), (10, 1, 15), (3, 1, 16), (6, 1, 16), (8, 1, 16), (10, 1, 16);`. Below the query editor, the 'Result Grid' shows the data inserted into the 'mesa' table.

mesa_id	capacidad	disponibilidad	restaurante_id
1	3	1	9
2	6	1	9
3	8	1	9
4	10	1	9
5	3	1	10
6	6	1	10
7	8	1	10
8	10	1	10
9	3	1	11
10	6	1	11
11	8	1	11
12	10	1	11
13	3	1	12
14	6	1	12


```
1 -- Inserción en tabla restaurante
2
3 INSERT INTO restaurante (nombre, calle, numero, tipo_de_comida)
4 VALUES ('Bohemia Vegan', 'Av Patria', 1275, 'Vegana');
```

Resultado

111 01:08:39 INSERT INTO restaurante (nombre, calle, numero, tipo_de_comida) VALUES ('Bohemia Vegan', 'Av Patria', 1... 1 row(s) affected

```
1 -- Inserción en tabla restaurante
2
3 INSERT INTO restaurante (nombre, calle, numero, tipo_de_comida)
4 VALUES ('Bohemia Vegan', 'Av Patria', 1275, 'Vegana');
5
6
7 SELECT * FROM RESTAURANTE;
```

	restaurante_id	nombre	calle	numero	tipo_de_comida
	1	Bohemia Mexican	Av Puevredon	1345	Mexicana
	2	Bohemia Chinese	Gral Paz	72	China
	3	Bohemia Arabian	Av 24 de Septiembre	791	Arabe
	4	Bohemia Burouer	Obispo Trejo	1200	Hamburguesa
	5	Bohemia Italian	Av Colon	2840	Italiana
	6	Bohemia Indian	Rondeu	439	India
	7	Bohemia Grill	Rafael Nuñez	3248	Parrilla
	8	Bohemia Fish & Shellfish	Sucre	2504	Pescado y Marisco
	9	Bohemia Vegan	Av Patria	1275	Vegana

MODIFICAR DATOS

UPDATE:

Con este sentencia, cambiaremos (actualizaremos) la capacidad del primer registro de la tabla mesa.

```
UPDATE mesa SET capacidad = 4 WHERE mesa_id = 1;
```

The screenshot shows a database management tool interface. On the left, a 'Navigator' pane displays a tree structure of schemas. The 'reserva_test' schema is expanded, showing a table named 'mesa'. The 'mesa' table is further expanded, showing its columns: 'mesa_id', 'capacidad', 'disponibilidad', and 'restaurante_id'. On the right, a 'Query 1' pane shows the following SQL queries:

```
1 • UPDATE mesa SET capacidad = 4 WHERE mesa_id = 1;
2 • select * from mesa;
```

Below the queries, a 'Result Grid' pane displays the results of the second query. The grid has four columns: 'mesa_id', 'capacidad', 'disponibilidad', and 'restaurante_id'. The data is as follows:

mesa_id	capacidad	disponibilidad	restaurante_id
1	4	1	9
2	6	1	9
3	8	1	9
4	10	1	9
5	3	1	10
6	6	1	10
7	8	1	10
8	10	1	10
9	3	1	11
10	6	1	11
11	8	1	11
12	10	1	11
13	3	1	12
14	6	1	12
15	8	1	12

Cambio de nombre de la calle (columna calle) del restaurante con Id= 9

```
9      -- Cambio de atributo en tabla Restaurante
10 •   UPDATE restaurante
11     SET calle = 'Av del Trabajo'
12     WHERE restaurante_id = 9
```

Resultado

115 01:21:42 UPDATE restaurante SET calle = 'Av del Trabajo' WHERE restaurante_id = 9

SQL Editor Interface:

```
-- Cambio de atributo en tabla Restaurante
UPDATE restaurante
SET calle = 'Av del Trabajo'
WHERE restaurante_id = 9
```

restaurante_id	nombre	calle	numero	tipo_de_comida
1	Bohemia Mexican	Av Puevrrredon	1345	Mexicana
2	Bohemia Chinese	Gral Paz	72	China
3	Bohemia Arabian	Av 24 de Setiembre	791	Arabe
4	Bohemia Burquer	Obisoo Trelo	1200	Hamburquesa
5	Bohemia Italian	Av Colon	2840	Italiana
6	Bohemia Indian	Rondeu	439	India
7	Bohemia Grill	Rafael Nuñez	3248	Parrilla
8	Bohemia Fish & Shellfish	Sucre	2504	Pescado v Marisco
9	Bohemia Vegan	Av del Trabajo	1275	Veana

ELIMINAR DATOS

DELETE:

Con esta sentencia,
eliminamos el registro 1 de la
tabla mesa.

```
DELETE FROM mesa  
WHERE mesa_id = 1 LIMIT 1;
```

The screenshot displays a database management interface. On the left, the 'Navigator' pane shows a tree structure of schemas. The 'reserva_test' schema is expanded, showing a table named 'mesa' with columns: 'mesa_id', 'capacidad', 'disponibilidad', and 'restaurante_id'. The 'Query 1' pane on the right contains the following SQL statements:

```
1 • DELETE FROM mesa WHERE mesa_id = 1;  
2 • select * from mesa;  
3
```

Below the query editor, the 'Result Grid' shows the data from the 'mesa' table after the deletion. The table has 16 rows, with the first row (mesa_id = 2) highlighted. The columns are 'mesa_id', 'capacidad', 'disponibilidad', and 'restaurante_id'.

	mesa_id	capacidad	disponibilidad	restaurante_id
▶	2	6	1	9
	3	8	1	9
	4	10	1	9
	5	3	1	10
	6	6	1	10
	7	8	1	10
	8	10	1	10
	9	3	1	11
	10	6	1	11
	11	8	1	11
	12	10	1	11
	13	3	1	12
	14	6	1	12
	15	8	1	12
	16	10	1	17

Eliminar la fila con Id=9 de la tabla Restaurante

```
1  
2 -- Eliminación de una fila de la tabla Restaurante  
3 • DELETE FROM restaurante  
4 WHERE restaurante_id= 9;
```

Resultado

123 01:35:02 DELETE FROM restaurante WHERE restaurante_id= 9

```
1  
2 -- Eliminación de una fila de la tabla Restaurante  
3 • DELETE FROM restaurante  
4 WHERE restaurante_id= 9;  
5  
6 • SELECT * FROM restaurante;
```

<

Result Grid | Filter Rows: | Edit: | Export/Import:

	restaurante_id	nombre	calle	numero	tipo_de_comida
	1	Bohemia Mexican	Av Puevrrredon	1345	Mexicana
	2	Bohemia Chinese	Gral Paz	72	China
	3	Bohemia Arabian	Av 24 de Setiembre	791	Arabe
	4	Bohemia Burouer	Obispo Treio	1200	Hamburguesa
	5	Bohemia Italian	Av Colon	2840	Italiana
	6	Bohemia Indian	Rondeu	439	India
	7	Bohemia Grill	Rafael Nuñez	3248	Parrilla
	8	Bohemia Fish & Shellfish	Sucre	2504	Pescado v Marisco
	NULL	NULL	NULL	NULL	NULL

UNA TABLA SOLA MOSTRANDO TODOS LOS DATOS

	reserva_id	fecha	hora	usuario_id	restaurante_id
▶	1	2024-06-13	08:00:00	1	2
	2	2024-06-13	09:00:00	2	3
	3	2024-06-13	10:00:00	3	4
	4	2024-06-13	11:00:00	4	5
	5	2024-06-13	12:00:00	5	2
	6	2024-06-13	13:00:00	6	4

UNA TABLA SOLA MOSTRANDO ALGUNAS COLUMNAS







```
1 select usuario_id, email
2 from usuarios
3
4
```

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

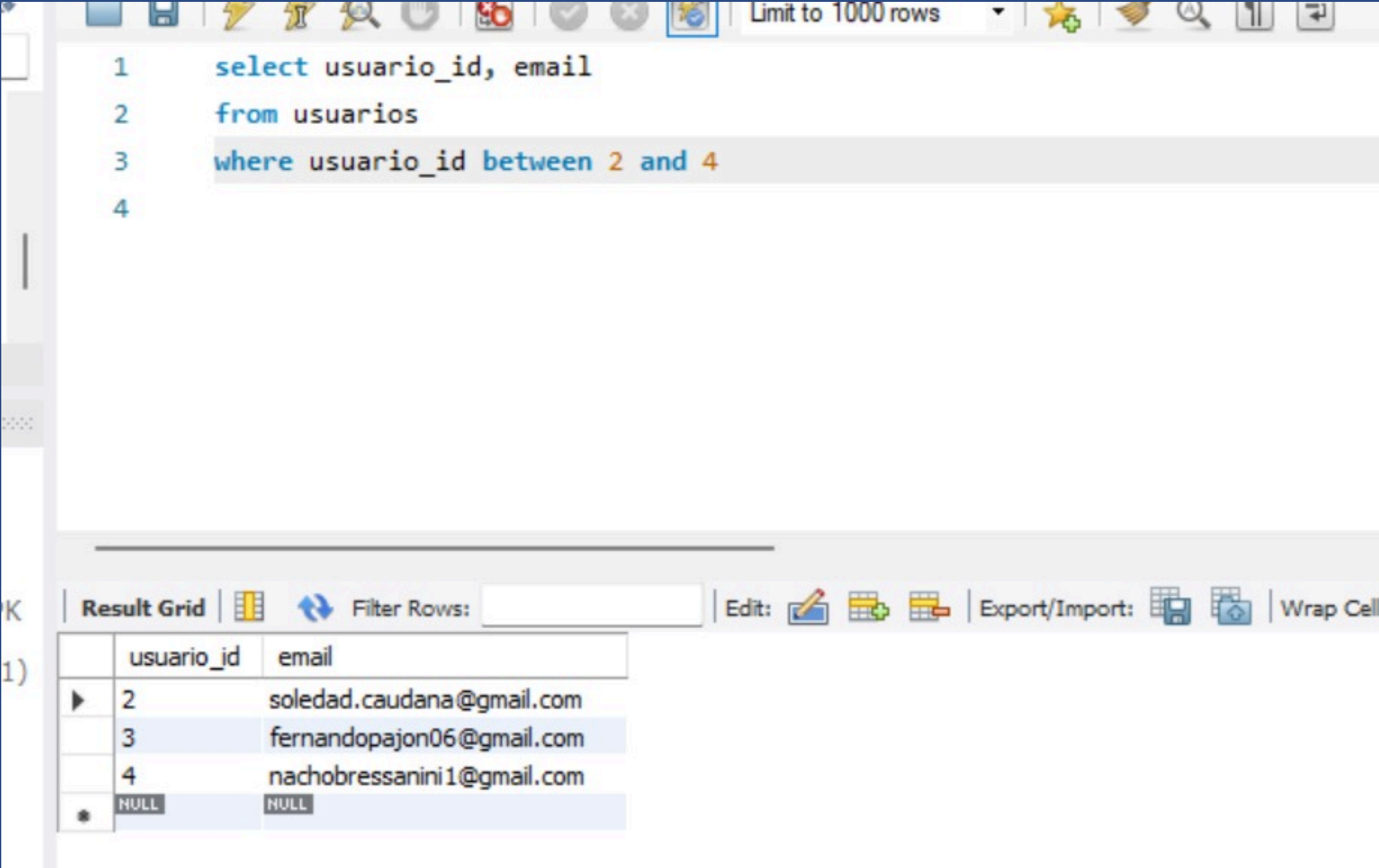
	usuario_id	email
▶	1	pabblloocanno@gmail.com
	2	soledad.caudana@gmail.com
	3	fernandopajon06@gmail.com
	4	nachobressanini1@gmail.com
★	NULL	NULL

UNA SOLA TABLA CON WHERE

```
1  select usuario_id, email
2  from usuarios
3  where usuario_id = 1
4
```

Result Grid			Filter Rows: <input type="text"/>	Edit: 			Export/Import: 
	usuario_id	email					
▶	1	pabbllloocanno@gmail.com					
✱	NULL	NULL					

UNA SOLA TABLA CON WHERE UTILIZANDO BETWEEN



The screenshot shows a database query editor interface. The top toolbar includes icons for saving, undo, redo, and other standard database operations, along with a "Limit to 1000 rows" dropdown. The SQL query is as follows:

```
1 select usuario_id, email
2 from usuarios
3 where usuario_id between 2 and 4
4
```

Below the query editor is the "Result Grid" section, which displays the results of the query. It includes a "Filter Rows:" input field and various action buttons like "Edit:", "Export/Import:", and "Wrap Cell". The results are shown in a table with two columns: "usuario_id" and "email".

	usuario_id	email
▶	2	soledad.caudana@gmail.com
	3	fernandopajon06@gmail.com
	4	nachobressanini1@gmail.com
★	NULL	NULL

UNA SOLA TABLA CON WHERE UTILIZANDO LIMIT

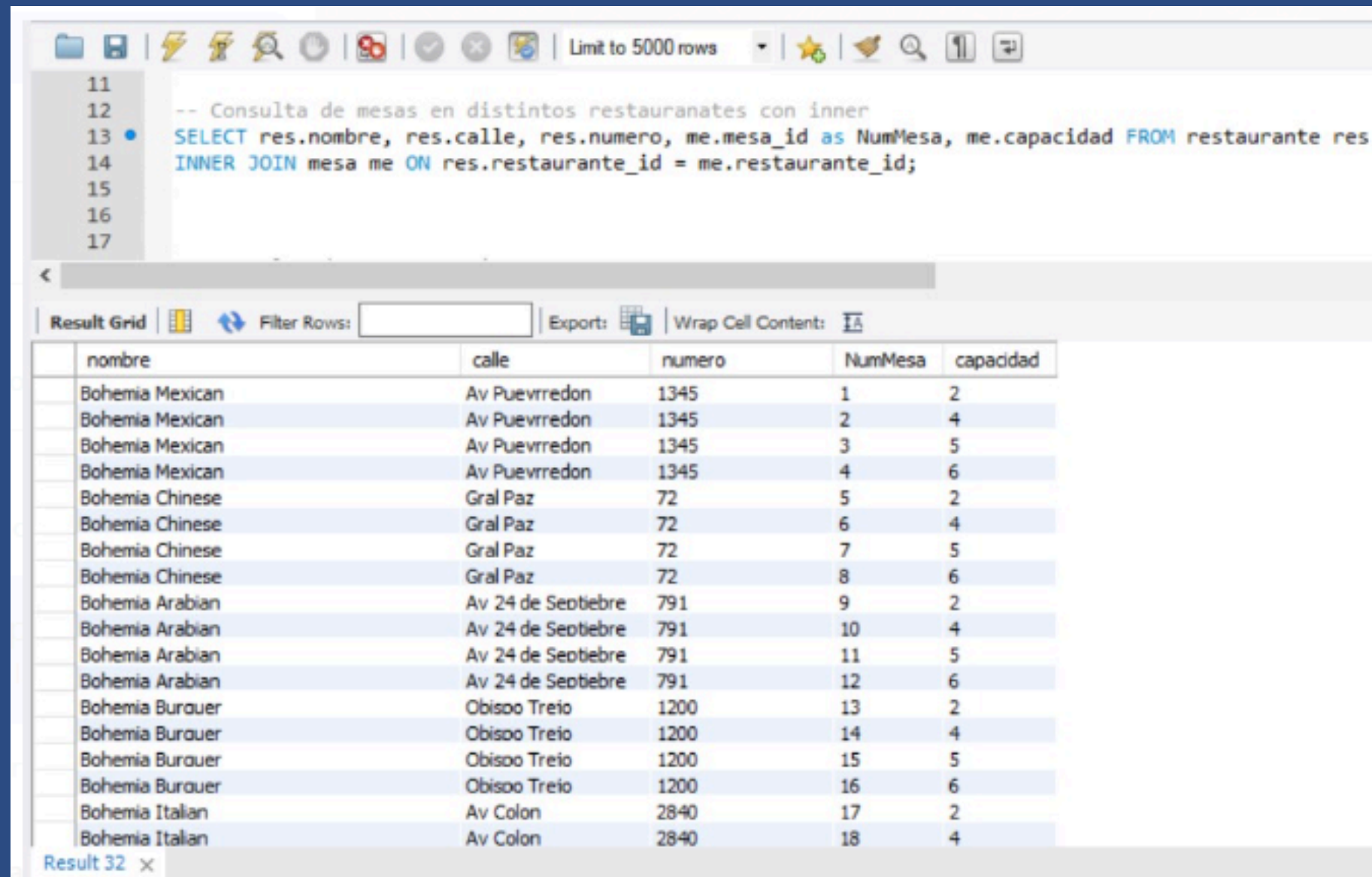
```
1  select usuario_id, email
2  from usuarios
3  where usuario_id limit 2
4
5
```

Result Grid | Filter Rows: | Edit: | Export/Import:

	usuario_id	email
▶	1	pabbloocanno@gmail.com
	2	soledad.caudana@gmail.com
✱	NULL	NULL

MAS DE UNA TABLA CON INNER JOIN

```
5
6 -- Selección de reservas con INNER JOIN
7 • SELECT res.reserva_id, us.apellido, us.nombre, r.fecha, r.hora, res.nombre,
8     res.tipo_de_comida as tipoComida, res.calle as Direccion from reserva r
9     INNER JOIN usuarios us ON r.usuario_id = us.usuario_id
10    INNER JOIN restaurante res ON r.restaurante_id = res.restaurante_id;
11
```



The screenshot shows a database query tool interface. At the top, there's a toolbar with various icons and a 'Limit to 5000 rows' dropdown. Below the toolbar, the SQL query is displayed in a text area. The query is an INNER JOIN between 'restaurante' and 'mesa' tables. Below the query, there's a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The main part of the grid is a table with 5 columns: 'nombre', 'calle', 'numero', 'NumMesa', and 'capacidad'. The table contains 18 rows of data, showing different restaurants and their respective tables. The status bar at the bottom indicates 'Result 32' and a close button.

nombre	calle	numero	NumMesa	capacidad
Bohemia Mexican	Av Puevrrredon	1345	1	2
Bohemia Mexican	Av Puevrrredon	1345	2	4
Bohemia Mexican	Av Puevrrredon	1345	3	5
Bohemia Mexican	Av Puevrrredon	1345	4	6
Bohemia Chinese	Gral Paz	72	5	2
Bohemia Chinese	Gral Paz	72	6	4
Bohemia Chinese	Gral Paz	72	7	5
Bohemia Chinese	Gral Paz	72	8	6
Bohemia Arabian	Av 24 de Seotiebre	791	9	2
Bohemia Arabian	Av 24 de Seotiebre	791	10	4
Bohemia Arabian	Av 24 de Seotiebre	791	11	5
Bohemia Arabian	Av 24 de Seotiebre	791	12	6
Bohemia Burouer	Obisoo Treio	1200	13	2
Bohemia Burouer	Obisoo Treio	1200	14	4
Bohemia Burouer	Obisoo Treio	1200	15	5
Bohemia Burouer	Obisoo Treio	1200	16	6
Bohemia Italian	Av Colon	2840	17	2
Bohemia Italian	Av Colon	2840	18	4

MAS DE UNA TABLA CON INNER JOIN Y CON FILTROS

136 02:03:39 SELECT r.reserva_id, us.apellido, us.nombre, r.fecha, r.hora, res.nombre, res.tipo_de_comida as tipoComida, re... 1 row(s) returned

Limit to 5000 rows

```
11
12 -- consulta de reserva unica con INNER JOIN y WHERE
13 • SELECT r.reserva_id, us.apellido, us.nombre, r.fecha, r.hora, res.nombre, res.tipo_de_comida as tipoComida,
14    res.calle as Direccion, res.numero FROM reserva r
15    INNER JOIN usuarios us ON r.usuario_id = us.usuario_id
16    INNER JOIN restaurante res ON r.restaurante_id = res.restaurante_id
17    WHERE r.reserva_id= 4;
18
```

Result Grid | Filter Rows: | Exports: | Wrap Cell Content:

	reserva_id	apellido	nombre	fecha	hora	nombre	tipoComida	Direccion	numero
	4	Bressanini	Ionado	2024-06-13	11:00:00	Bohemia Italian	Italiana	Av Colon	2840



¡MUCHAS GRACIAS!