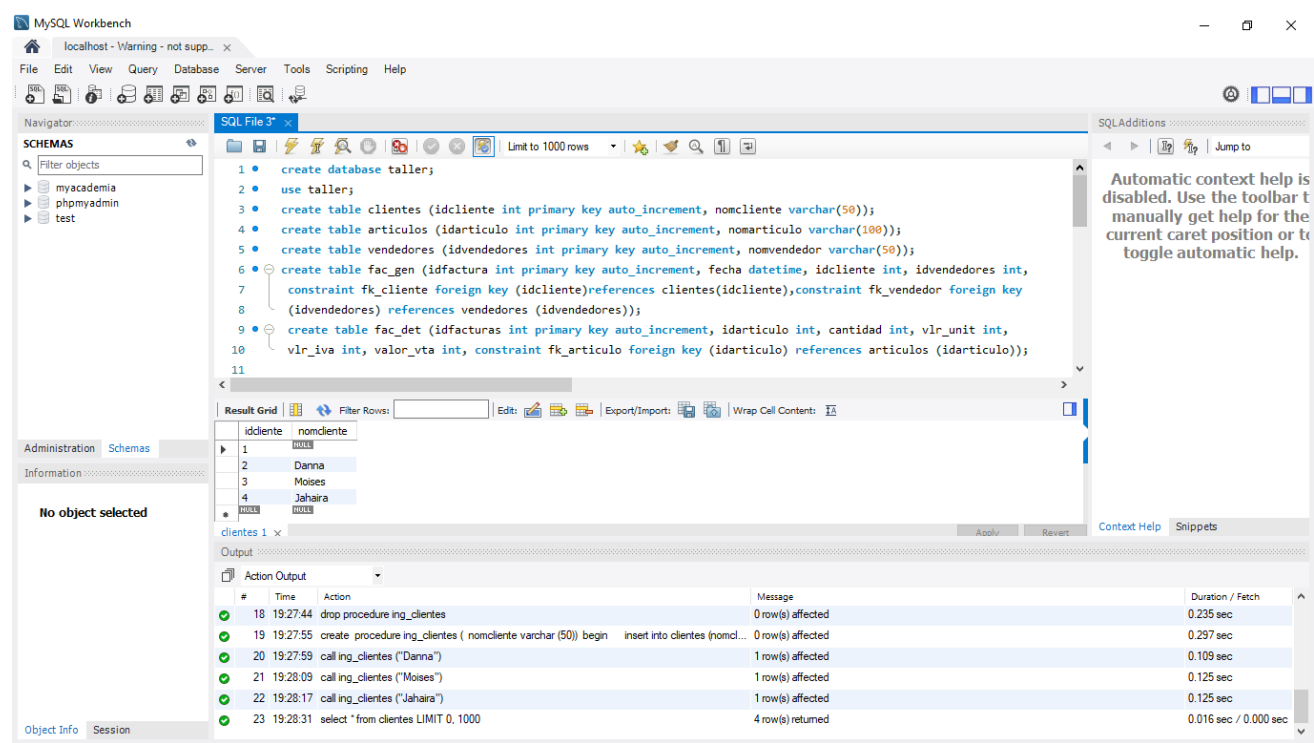
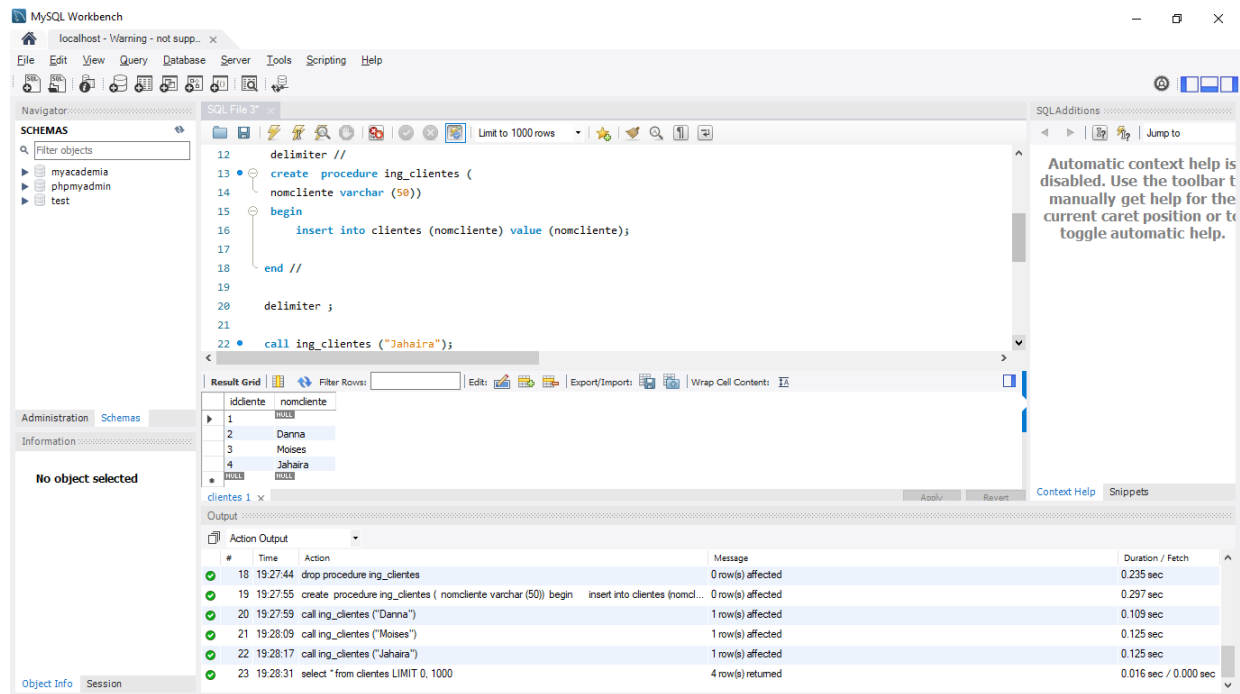


Base de datos – Moises Zabaleta, Danna Beltran, Jahaira Palencia

1. Base de datos



2. Procedimientos almacenados



MySQL Workbench

localhost - Warning - not supp...

FileEditViewQueryDatabaseServerToolsScriptingHelp

Navigator

SQL File 3*

Limit to 1000 rows

SCHEMAS

Filter objects

myacademia

phpmyadmin

test

Administration

Schemas

Information

No object selected

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

delimiter //

create procedure ing_articulos (

nomarticulo varchar (100))

begin

insert into articulos (nomarticulo) value (nomarticulo);

end //

delimiter ;

call ing_articulos ("tablet");

select * from articulos;

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

idarticulo

nomarticulo

1

2

3

4

total

1000

1000

1000

1000

1000

1000

articulos 6 x

Apply

Revert

Context Help

Snippets

Output

Action Output

#

Time

Action

Message

Duration / Fetch

MySQL Workbench

localhost - Warning - not supp...

FileEditViewQueryDatabaseServerToolsScriptingHelp

Navigator

SQL File 3*

Limit to 1000 rows

SCHEMAS

Filter objects

myacademia

phpmyadmin

test

Administration

Schemas

Information

No object selected

46

47

48

49

50

51

52

53

54

55

56

57

58

delimiter //

create procedure ing_vendedor (

nomvendedor varchar (100))

begin

insert into vendedores (nomvendedor) value (nomvendedor);

end //

delimiter ;

call ing_vendedor ("Stefania");

select * from vendedores;

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

idvendedores

nomvendedor

3

4

5

total

1000

1000

1000

1000

1000

1000

vendedores 10 x

Apply

Revert

Context Help

Snippets

Output

Action Output

#

Time

Action

Message

Duration / Fetch

48

19:52:44

delete from vendedores where idvendedores =1

1 row(s) affected

0.328 sec

49

19:52:55

delete from vendedores where idvendedores =2

1 row(s) affected

0.141 sec

50

19:53:00

select * from vendedores LIMIT 0, 1000

3 row(s) returned

0.031 sec / 0.000 sec

3. Procedimiento almacenado en las tablas fac_gen y fac_det

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- banco
- bdtest
- clase_test
- prueba1
- sakila
- sys
- taller_8_mayo
- taller_final
 - Tables
 - articulos
 - clientes
 - fac_det
 - fac_gen
 - vendedores
 - Views
 - Stored Procedures

Administration Schemas

Information

Schema: taller_final

SQL File 3* SQL File 4* SQL File 8* libreria dana taller final final no da mas*

Limit to 1000 rows

```
76
77 delimiter ;
78
79 • call ing_fac_gen (1,1);
80
81 • select * from fac_gen;
```

Result Grid

	idfactura	fecha	idcliente	idvendedores
▶	1	2024-05-22 17:11:35	1	1
	2	2024-05-22 17:11:42	2	1
	4	2024-05-22 17:11:58	3	1
	5	2024-05-22 17:47:23	1	1
•	NULL	NULL	NULL	NULL

fac_gen 1 x

Output

Action Output

#	Time	Action	Message
✗	6 17:45:28	call ing_fac_gen (2,1)	Error Code: 1146. Table 'bdtest.fac_gen' doesn't exist
✗	7 17:46:15	call ing_fac_gen (2,1)	Error Code: 1146. Table 'bdtest.fac_gen' doesn't exist
✗	8 17:46:47	select * from fac_gen LIMIT 0, 1000	Error Code: 1146. Table 'bdtest.fac_gen' doesn't exist
✗	9 17:46:58	create database taller_final	Error Code: 1007. Can't create database 'taller_final'; database exists
✓	10 17:47:23	call ing_fac_gen (1,1)	1 row(s) affected
✓	11 17:47:28	select * from fac_gen LIMIT 0, 1000	4 row(s) returned

Object Info Session

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- banco
- bdtest
- clase_test
- prueba1
- sakila
- sys
- taller_8_mayo
- taller_final
 - Tables
 - articulos
 - clientes
 - fac_det
 - fac_gen
 - vendedores
 - Views
 - Stored Procedures

Administration Schemas

Information

Table: fac_det

Columns:

- idfacturas int AI PK
- idarticulo int
- cantidad int
- vlr_unit int
- vlr_iva int
- valor_vta int

SQL File 3* SQL File 4* SQL File 8* libreria dana taller final final no da mas*

Limit to 1000 rows

```
89 cantidad int,
90 vlr_unit int
91 )
92 begin
93     insert into fac_det (idarticulo,cantidad,vlr_unit,vlr_iva,valor_vta) value (id_articulo,cantidad,vlr_unit,vlr_unit*0.19,vlr_unit
94 end //
95
96
97 delimiter ;
98 • call ing_fac_det (2,15,100000);
99 • select * from fac_det;
```

Result Grid

	idfacturas	idarticulo	cantidad	vlr_unit	vlr_iva	valor_vta
▶	1	1	10	150000	28500	178500
	2	2	15	100000	19000	119000
•	NULL	NULL	NULL	NULL	NULL	NULL

fac_det 3 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓	13 17:53:48	call ing_fac_det (1,10,150000)	1 row(s) affected	0.015 sec

Object Info Session

4. FUNCION SUM()

a. procedimiento permita ver el total de las ventas

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

banco

bdtest

clase_test

prueba1

sakila

sys

taller_8_mayo

taller_final

Tables

articulos

clientes

fac_det

fac_gen

vendedores

Views

Stored Procedures

Administration Schemas

Information

Table: fac_det

Columns:

idfacturas

idarticulo

cantidad

vir_unit

vir_iva

valor_vta

int AI PK

int

int

int

int

int

SQL File 3* SQL File 4* SQL File 8* libreria dana taller final final no da mas*

Limit to 1000 rows

112

113 DELIMITER //

114

115 • CREATE PROCEDURE TotalVentas()

116 BEGIN

117 SELECT SUM(valor_vta) AS TotalDeVentas FROM fac_det;

118 END //

119

120 DELIMITER ;

121 • CALL TotalVentas();

122

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

TotalDeVentas

416500

b. procedimiento permita ver el total de las ventas por cliente

Limit to 1000 rows

130 • CREATE PROCEDURE TotalVentas_Cliente(
131 in ing_idcliente INT
132)
133 BEGIN
134 SELECT
135 idcliente,
136 SUM(fac_det.valor_vta) AS TotalDeVentas
137 FROM
138 fac_gen
139 JOIN
140 fac_det ON idfactura = idfactura
141 WHERE
142 fac_gen.idcliente = ing_idcliente
143 GROUP BY
144 fac_gen.idcliente;
145 END //
146 DELIMITER ;

Result Grid

Filter Rows:

Export:

Wrap Cell Conte

idcliente

TotalDeVentas

2

2082500

c. Crear un procedimiento permita ver el total de las ventas por vendedor

The screenshot displays a SQL development environment. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The main editor shows a PL/SQL procedure named 'TotalVentas_vendedor' with the following code:

```
154 DELIMITER //  
155 CREATE PROCEDURE TotalVentas_vendedor(  
156     in ing_idvendedores INT  
157 )  
158 BEGIN  
159     SELECT  
160         idvendedores,  
161         SUM(fac_det.valor_vta) AS TotalDeVentas  
162     FROM  
163         fac_gen  
164     JOIN  
165         fac_det ON idfactura = idfactura  
166     WHERE  
167         fac_gen.idvendedores = ing_idvendedores  
168     GROUP BY  
169         fac_gen.idvendedores;  
170 END //  
171 DELIMITER ;
```

Below the editor, the 'Result Grid' tab is active, showing the output of the procedure. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result table has two columns: 'idvendedores' and 'TotalDeVentas'.

idvendedores	TotalDeVentas
2	833000

The status bar at the bottom indicates 'Result 18'.

5. FUNCION AVG()

- a. Crear un procedimiento permita ver el promedio de las ventas

The screenshot shows a SQL IDE with a toolbar at the top. The main editor displays SQL code for creating and calling a stored procedure. The code is as follows:

```
177
178
179  -----
180  DELIMITER //
181  • CREATE PROCEDURE promedioVentas()
182  BEGIN
183      SELECT AVG(valor_vta) AS PromedioDeVentas FROM fac_det;
184  END //
185
186  DELIMITER ;
187
188  • CALL promedioVentas();
189
```

Below the code editor, there is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid shows a single row with the following data:

TotalDeVentas
133875.0000

- b. Crear un procedimiento permita ver el promedio de las ventas por cliente

The screenshot shows a SQL IDE with a toolbar at the top. The main editor displays SQL code for creating and calling a stored procedure. The code is as follows:

```
196  DELIMITER //
197  • CREATE PROCEDURE promedioVentas_Cliente(
198      in ing_idcliente INT
199  )
200  BEGIN
201      SELECT
202          idcliente,
203          AVG(fac_det.valor_vta) AS PromedioDeVentas
204      FROM
205          fac_gen
206      JOIN
207          fac_det ON idfactura = idfactura
208      WHERE
209          fac_gen.idcliente = ing_idcliente
210      GROUP BY
211          fac_gen.idcliente;
212  END //
```

Below the code editor, there is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid shows a single row with the following data:

idcliente	TotalDeVentas
3	43987.5000

