

Descripción: Personas que se registran para comprar productos en la tienda en línea.

Campos:

id_cliente (PK) → Identificador único del cliente.

nombre → Nombre completo del cliente.

correo → Correo electrónico del cliente.

Entidad 2: Productos

Descripción: Artículos que están disponibles para que los clientes los compren.

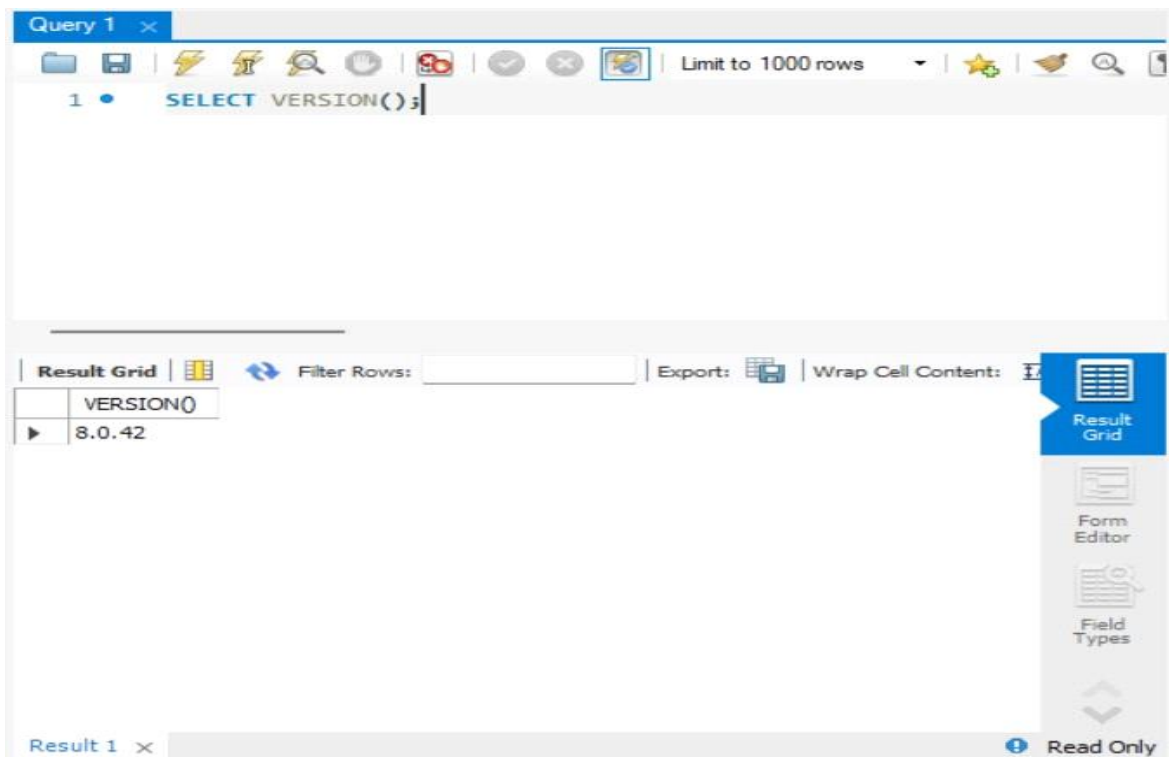
Campos:

id_producto (PK) → Identificador único del producto.

nombre_producto → Nombre del producto.

precio → Precio de venta del producto.

taller 2.2



Action Output			
	#	Time	Action
✓	1	09:57:21	SELECT VERSION() LIMIT 0, 1000
✓	2	09:59:42	CREATE DATABASE TallerBD

1 **SHOW DATABASES;**

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Database

▶ information_schema

mysql

performance_schema

sakila

sys

tallerbd

world

Result Grid

Form Editor

Field Types

punto 3.1.2

Query 1 x

Limit to 1000 rows

```

27 ) ENGINE=InnoDB;
28
29 -- Tabla Profesores
30 CREATE TABLE IF NOT EXISTS Profesores (
31     id_profesor INT AUTO_INCREMENT PRIMARY KEY,
32     nombre VARCHAR(100) NOT NULL,
33     especialidad VARCHAR(100),
34     correo VARCHAR(150) UNIQUE,
35     telefono VARCHAR(15),
36     fecha_contratacion DATE
37 ) ENGINE=InnoDB;
38
39 SHOW TABLES;

```

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

Tables_in_tallerbd
profesores
estudiantes
cursos

Result 3 x

Output

Action Output

#	Time	Action	Message
5	10:38:29	CREATE DATABASE IF NOT EXISTS TallerBD CHARACTER SET utf8mb4 COLLATE utf8mb4_spanish_ci	1 row(s) affected, 1 warning
6	10:38:29	USE TallerBD	0 row(s) affected
7	10:38:29	CREATE TABLE IF NOT EXISTS Estudiantes (id_estudiante INT AUTO_INCREMENT PRIMARY KEY, -- P...	0 row(s) affected
8	10:38:30	CREATE TABLE IF NOT EXISTS Cursos (id_curso INT AUTO_INCREMENT PRIMARY KEY, nombre VAR...	0 row(s) affected
9	10:38:30	CREATE TABLE IF NOT EXISTS Profesores (id_profesor INT AUTO_INCREMENT PRIMARY KEY, nomb...	0 row(s) affected
10	10:38:30	SHOW TABLES	3 row(s) returned

punto 3.1.4

```
1 • SELECT nombre
2 FROM Estudiantes
3 WHERE telefono IS NOT NULL;
```

Result Grid





Filter Rows:

Export:



nombre
María González
Luisa Martínez

```
1 • SELECT nombre, credits
2 FROM Cursos
3 WHERE credits >= 4;
```

Result Grid   Filter Rows: | Ex

	nombre	credits
▶	Matemáticas I	4
	Programación	5

punto 3.1.5




1.

```
1 • SELECT nombre
2 FROM Profesores
3 ORDER BY nombre ASC;
```

Result Grid		Filter Rows:	Exp
	nombre		
▶	Ana Ramírez		
	Carlos Díaz		
	Marina López		





2.

```
1 • SELECT COUNT(*) AS prof_contratados_desde_2020
2 FROM Profesores
3 WHERE fecha_contratacion > '2020-01-01';
```

Result Grid				Filter Rows: <input type="text"/>	Export: 	Wrap Cell
	prof_contratados_desde_2020					
2						

3.

```
1 • SELECT especialidad, COUNT(*) AS cantidad
2 FROM Profesores
3 GROUP BY especialidad;
4 |
```

Result Grid		 Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	especialidad	cantidad		
▶	Matemáticas	1		
	Física	1		
	Informática	1		

punto 3.1.6

MySQL Workbench

Local Taller x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

sys

tallerbd

Tables

Views

Stored Procedures

Functions

Administration Schemas

Information

Query 1 Query1*

Limit to 1000 rows

```
-- 1. Listar estudiantes ordenados por fecha de nacimiento descendente (del más joven al más viejo)
1  SELECT nombre, fecha_nacimiento
2  FROM Estudiantes
3
4  ORDER BY fecha_nacimiento DESC;
5
6  SELECT COUNT(*) AS total_estudiantes
7  FROM Estudiantes;
8
9  SELECT COUNT(*) AS cursos_mayor_d_credits
10 FROM Cursos
11 WHERE credits >= 4;
12
13 SELECT AVG(credits) AS promedio_credits
14 FROM Cursos;
15
16 SELECT credits, COUNT(*) AS cant_cursos
17 FROM Cursos
18 GROUP BY credits;
```

Schema: tallerbd

Result Grid

credits	cant_cursos
4	1
3	1
5	1

Estudiantes 11 Result 12 Result 13 Result 14 Result 15 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
27	14:08:44	SELECT AVG(credits) AS promedio_credits FROM Cursos LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
28	14:09:53	SELECT nombre, fecha_nacimiento FROM Estudiantes ORDER BY fecha_nacimiento DESC LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
29	14:09:53	SELECT COUNT(*) AS total_estudiantes FROM Estudiantes LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
30	14:09:53	SELECT COUNT(*) AS cursos_mayor_d_credits FROM Cursos WHERE credits >= 4 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
31	14:09:53	SELECT AVG(credits) AS promedio_credits FROM Cursos LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
32	14:09:53	SELECT credits, COUNT(*) AS cant_cursos FROM Cursos GROUP BY credits LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

2:10 p. m.
31/05/2025

3.2.1

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with 'sys' selected. The main editor shows a SQL script with the following queries:

```
1 * INSERT INTO Matriculas (id_estudiante, id_curso, fecha_matricula, nota_final)
2 VALUES
3 (1, 1, '2024-03-05', 8.75),
4 (2, 2, '2024-03-10', 9.00),
5 (3, 3, '2024-04-01', NULL); -- Nota pendiente
```

The bottom panel shows the 'Output' tab with the following results:

Time	Action	Message	Duration / Fecha
55 14:45:03	USE Table0	0 rows(s) affected	0.000 sec
56 14:45:03	CREATE TABLE IF NOT EXISTS Estudiantes (id_estudiante INT AUTO_INCREMENT PRIMARY KEY, -- PK autoncremental nombre VARCHAR(100) NOT NULL, credits INT N...	0 rows(s) affected, 1 warning(s): 1050 Table 'estudiantes' already exists	0.046 sec
57 14:45:03	CREATE TABLE IF NOT EXISTS Cursos (id_curso INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(100) NOT NULL, credits INT N...	0 rows(s) affected, 1 warning(s): 1050 Table 'cursos' already exists	0.016 sec
58 14:45:03	CREATE TABLE IF NOT EXISTS Profesores (id_profesor INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(100) NOT NULL, especial...	0 rows(s) affected, 1 warning(s): 1050 Table 'profesores' already exists	0.015 sec
59 14:45:03	CREATE TABLE IF NOT EXISTS Matriculas (id_matricula INT AUTO_INCREMENT PRIMARY KEY, id_estudiante INT NOT NULL, id_curso INT...	0 rows(s) affected	1.100 sec
60 14:45:01	INSERT INTO Matriculas (id_estudiante, id_curso, fecha_matricula, nota_final) VALUES (1, 1, '2024-03-05', 8.75), (2, 2, '2024-03-10', 9.00), (3, 3, 2...	3 rows(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.062 sec

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with 'sys' selected. The main editor shows a SQL script with the following query:

```
1 * SELECT
2 e.id_estudiante,
3 e.nombre AS nombre_estudiante,
4 c.nombre AS nombre_curso,
5 m.fecha_matricula
6 FROM Estudiantes e
7 LEFT JOIN Matriculas m
8 ON e.id_estudiante = m.id_estudiante
9 LEFT JOIN Cursos c
10 ON m.id_curso = c.id_curso
```

The bottom panel shows the 'Result Grid' with the following results:

id_estudiante	nombre_estudiante	nombre_curso	fecha_matricula
1	María González	Matemáticas I	2024-03-05 00:00:00
2	Juan Pérez	Física I	2024-03-10 00:00:00
3	Luisa Martínez	Programación	2024-04-01 00:00:00

The bottom panel also shows the 'Output' tab with the following results:

Time	Action	Message	Duration / Fecha
57 14:45:03	CREATE TABLE IF NOT EXISTS Cursos (id_curso INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(100) NOT NULL, credits INT N...	0 rows(s) affected, 1 warning(s): 1050 Table 'cursos' already exists	0.016 sec
58 14:45:03	CREATE TABLE IF NOT EXISTS Profesores (id_profesor INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(100) NOT NULL, especial...	0 rows(s) affected, 1 warning(s): 1050 Table 'profesores' already exists	0.015 sec
59 14:45:03	CREATE TABLE IF NOT EXISTS Matriculas (id_matricula INT AUTO_INCREMENT PRIMARY KEY, id_estudiante INT NOT NULL, id_curso INT...	0 rows(s) affected	1.100 sec
60 14:45:01	INSERT INTO Matriculas (id_estudiante, id_curso, fecha_matricula, nota_final) VALUES (1, 1, '2024-03-05', 8.75), (2, 2, '2024-03-10', 9.00), (3, 3, 2...	3 rows(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.062 sec
61 14:45:01	SELECT e.id_estudiante, e.nombre AS nombre_estudiante, c.nombre AS nombre_curso, m.fecha_matricula FROM Estudiantes e LEFT JOIN Mat...	3 rows(s) returned	0.000 sec / 0.000 sec
62 14:45:00	SELECT e.id_curso, c.nombre AS nombre_curso, e.nombre AS nombre_estudiante, m.fecha_matricula FROM Estudiantes e RIGHT JOIN Matricu...	3 rows(s) returned	0.000 sec / 0.000 sec



Navigator

SCHEMAS

Filter objects

- sys
- tallerdb
 - Tables
 - cursos
 - estudiantes
 - matriculas
 - profesores
 - Views
 - Stored Procedures
 - Functions

Administration Schemas

Information

Schema: sys

Query 1 Query1* Query1.* Query Query. Query2 x

Limit to 1000 rows

```
1 • SELECT
2   c.id_curso,
3   c.nombre AS nombre_curso,
4   e.nombre AS nombre_estudiante,
5   m.fecha_matricula
6 FROM Estudiantes e
7 RIGHT JOIN Matriculas m
8   ON e.id_estudiante = m.id_estudiante
9 RIGHT JOIN Cursos c
10  ON m.id_curso = c.id_curso;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	id_curso	nombre_curso	nombre_estudiante	fecha_matricula
▶ 1		Matemáticas I	María González	2024-03-05 00:00:00
2		Física I	Juan Pérez	2024-03-10 00:00:00
3		Programación	Luisa Martínez	2024-04-01 00:00:00

Result 1 x

Output

Action Output

#	Time	Action	Mes
57	14:45:03	CREATE TABLE IF NOT EXISTS Cursos (id_curso INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(100) NOT NULL, credits INT N...	0 ro
58	14:45:03	CREATE TABLE IF NOT EXISTS Profesores (id_profesor INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(100) NOT NULL, especial...	0 ro
59	14:45:03	CREATE TABLE IF NOT EXISTS Matriculas (id_matricula INT AUTO_INCREMENT PRIMARY KEY, id_estudiante INT NOT NULL, id_curso INT ...	0 ro
60	14:45:41	INSERT INTO Matriculas (id_estudiante, id_curso, fecha_matricula, nota_final) VALUES (1, 1, '2024-03-05', 8.75), (2, 2, '2024-03-10', 9.00), (3, 3, '2...	3 ro
61	14:48:21	SELECT e.id_estudiante, e.nombre AS nombre_estudiante, c.nombre AS nombre_curso, m.fecha_matricula FROM Estudiantes e LEFT JOIN Matr...	3 ro
62	14:48:50	SELECT c.id_curso, c.nombre AS nombre_curso, e.nombre AS nombre_estudiante, m.fecha_matricula FROM Estudiantes e RIGHT JOIN Matricul...	3 ro

Object Info Session



Q Buscar

