

Semana 4: Proyecto inicial y normalización

Objetivo: Aplicar normalización hasta 3NF, diseñar un diagrama ER y construir una base de datos para una tienda online (tablas: clientes, productos, pedidos).

Total horas: 20 horas.

Entregable final: Diagrama ER y base de datos normalizada de la tienda online en PostgreSQL, subida a GitHub con documentación en el README.

Recursos: PostgreSQL, DBeaver, VS Code (con SQLTools), Draw.io, GitHub, “SQL Performance Explained” (cap. 1), YouTube.

Distribución diaria (Lunes a Domingo)

Lunes: Repaso de normalización (3 horas)

- Temas: Normalización (1NF, 2NF, 3NF), conceptos de diseño relacional.
- Tareas:
 - Leer: “SQL Performance Explained” (cap. 1, ~20 páginas, ~60 min). Concéntrate en

normalización y ejemplos prácticos.

- Ver video: “Database Normalization - 1NF, 2NF, 3NF” (YouTube, canal Simplilearn, ~30 min)

(~30 min).

- Práctica:
 - Analiza la base de datos de la biblioteca (biblioteca_db) de la Semana 3. Verifica si cumple

1NF, 2NF y 3NF. Escribe un resumen (1 página en Google Docs o Notas) explicando cómo las tablas (libros, usuarios, prestamos) cumplen cada forma normal (~90 min).

- Entregable: Resumen de normalización de biblioteca_db en un documento.
- Horas: 3 horas.

Martes: Introducción a diagramas ER (3 horas)

- Temas: Modelado Entidad-Relación (ER), entidades, atributos, relaciones.
- Tareas:
 - Ver tutorial: “How to Create an ER Diagram” (YouTube, canal Lucidchart, ~20 min) (~20 min).
 - Explorar Draw.io (app.diagrams.net) y aprender a crear entidades y relaciones (~30 min).
- Práctica:
 - Diseña un diagrama ER preliminar para la tienda online con 3 entidades:

■ clientes (atributos: id, nombre, email).

■ productos (atributos: id, nombre, precio).

■ pedidos (atributos: id, cliente_id, producto_id, fecha).

■ Define relaciones (ej. un cliente puede tener muchos pedidos, un pedido incluye un producto) (~130 min).

- Entregable: Diagrama ER preliminar guardado como imagen (PNG/JPG) en Draw.io.
- Horas: 3 horas.

Miércoles: Diseño de la base de datos de la tienda online (3 horas)

- Temas: Traducir diagrama ER a tablas, claves primarias y foráneas.
- Tareas:
 - Ver video: “From ER Diagram to Database Schema” (YouTube, canal Database Star, ~20 min)

(~20 min).

- Revisa tu diagrama ER de la tienda online y asegúrate de que las relaciones sean claras (~30

min).

- Práctica:
 - Escribe un script SQL para crear la base de datos en PostgreSQL:

■ Crea la base: CREATE DATABASE tienda_online; (~10 min).

- Crea las tablas:

CREATE TABLE clientes (

- id SERIAL PRIMARY KEY,
- nombre VARCHAR(100),
- email VARCHAR(100)
-);

CREATE TABLE productos (

- id SERIAL PRIMARY KEY,
- nombre VARCHAR(100),
- precio NUMERIC
-);

CREATE TABLE pedidos (

- id SERIAL PRIMARY KEY,
- cliente_id INTEGER REFERENCES clientes(id),
- producto_id INTEGER REFERENCES productos(id),
- fecha DATE
-);

- "" (~100 min).

■

- Entregable: Script .sql con la creación de tienda_online (guardado como semana4_tienda.sql).

- Horas: 3 horas.

Jueves: Normalización de la tienda online (3 horas)

- Temas: Aplicar 1NF, 2NF, 3NF a la base de datos.
- Tareas:
 - Leer: Guía de normalización en “SQL Performance Explained” (sección sobre normalización, ~20 páginas, ~60 min).

- Práctica:

- Analiza la base de datos tienda_online:

- Verifica 1NF: Asegúrate de que no hay datos repetidos ni multivaluados (ej. el campo email en clientes debe ser único).

- Verifica 2NF: Confirma que no hay dependencias parciales (ej. precio en productos depende solo de id).

- Verifica 3NF: Elimina dependencias transitivas (ej. no incluir datos redundantes como el nombre del cliente en pedidos).

- Escribe un resumen (1 página) explicando cómo tienda_online cumple 3NF (~90 min).

- Ajusta el script SQL si es necesario (ej. añade restricciones UNIQUE o NOT NULL) (~30 min).

- Entregable: Resumen de normalización y script SQL actualizado (semana4_tienda.sql).
- Horas: 3 horas.

Viernes: Poblar la base de datos (2.5 horas)

- Temas: Insertar datos normalizados, consultas básicas.

- Tareas:

- Ver video: “Inserting Data in PostgreSQL” (YouTube, canal Database Star, ~15 min) (~15

min).

- Práctica:
 - Inserta 10 registros de prueba en tienda_online:
 - 5 clientes: INSERT INTO clientes (nombre, email) VALUES ('Ana', 'ana@example.com'), ...;
 - 3 productos: INSERT INTO productos (nombre, precio) VALUES ('Laptop', 999.99), ...;
 - 2 pedidos: INSERT INTO pedidos (cliente_id, producto_id, fecha) VALUES (1, 1, '2025-08-04'), ...; (~90 min).
 - Escribe 2 consultas SELECT para verificar los datos:
 - Ej. 1: SELECT * FROM clientes WHERE email LIKE '%@example.com';
 - Ej. 2: SELECT c.nombre, p.nombre FROM clientes c JOIN pedidos pe ON c.id = pe.cliente_id JOIN productos p ON pe.producto_id = p.id; (~45 min).
- Entregable: Script .sql con inserciones y 2 consultas, añadido a semana4_tienda.sql.
- Horas: 2.5 horas.

Sábado: Proyecto práctico y diagrama final (5 horas)

- Temas: Finalizar el diseño ER y la base de datos.
- Tareas:
 - Revisa y mejora el diagrama ER en Draw.io, asegurándote de que refleje las tablas

normalizadas y las relaciones correctas (~60 min).

- Exporta el diagrama como PNG/JPG y guárdalo en tu carpeta de proyecto (~15 min).
- Práctica:
 - Añade 5 registros más a cada tabla de tienda_online para tener un dataset más robusto (~60

min).

- Escribe 3 consultas adicionales para reportes:
 - Ej. 1: SELECT c.nombre, COUNT(pe.id) AS total_pedidos FROM clientes c LEFT JOIN pedidos pe ON c.id = pe.cliente_id GROUP BY c.nombre;
 - Ej. 2: SELECT p.nombre, p.precio FROM productos p WHERE p.precio > (SELECT AVG(precio) FROM productos);
 - Ej. 3: SELECT c.nombre, pe.fecha FROM clientes c JOIN pedidos pe ON c.id = pe.cliente_id WHERE pe.fecha >= '2025-08-01'; (~120 min).
- Entregable: Diagrama ER final (PNG/JPG), script .sql actualizado con 15 registros y 5 consultas totales.
- Horas: 5 horas.

Domingo: Documentación y GitHub (3.5 horas)

- Temas: Documentar el proyecto, actualizar GitHub.
- Tareas:
 - Revisa todas las consultas y la estructura de tienda_online en DBeaver para asegurar que

funcionan correctamente (~30 min).

- Actualiza tu repositorio GitHub:
 - Añade el archivo semana4_tienda.sql con la creación de la base, inserciones y consultas (~30 min).
 - Sube el diagrama ER (PNG/JPG) al repositorio (~15 min).
 - Actualiza el README.md con una sección para la Semana 4, incluyendo:
 - Descripción de la base de datos tienda_online (tablas, relaciones, normalización).
 - Explicación del diagrama ER.

- Lista de las 5 consultas con su propósito (ej. “Conteo de pedidos por cliente”).
- Resumen de lo aprendido (normalización, diagramas ER) (~120 min).
- Práctica: Escribe una consulta adicional para probar la base de datos (ej. `SELECT p.nombre, SUM(p.precio) AS total FROM productos p JOIN pedidos pe ON p.id = pe.producto_id GROUP BY p.nombre;`) (~30 min).
- Entregable: Repositorio GitHub actualizado con script SQL, diagrama ER y README.
- Horas: 3.5 horas.

Resumen de la Semana 4

- Total horas: 20 horas (Lunes: 3, Martes: 3, Miércoles: 3, Jueves: 3, Viernes: 2.5, Sábado: 5, Domingo: 3.5).
- Entregables:
 - Resumen de normalización de biblioteca_db (1 página).
 - Diagrama ER final de tienda_online (PNG/JPG).
 - Script semana4_tienda.sql con creación de tienda_online, 15 registros y 5 consultas SQL.
 - Repositorio GitHub actualizado con script, diagrama y README detallado.
- Notas:
 - Usa PostgreSQL local en tu MacBook para tienda_online, manteniendo el uso de RAM bajo.
 - Si tienes problemas con Draw.io o normalización, consulta tutoriales en YouTube o la documentación de PostgreSQL.
 - Guarda los scripts y diagramas en una carpeta (ej. /Documentos/BasesDatos/Semana4) y respáldala en Google Drive.

Instrucciones adicionales

- Seguimiento: Registra el tiempo y tareas completadas en tu documento de Google Docs o Notas para mantener el control. Añade esta distribución a tu calendario.
 - PDF: Copia este texto y pégalo en el documento de tu calendario (Google Docs/Word) para incluirlo como parte de la Semana 4. Usa viñetas y encabezados claros antes de exportar a PDF.
 - Soporte: Si necesitas ayuda con algún paso (ej. crear el diagrama ER, normalizar la base de datos, o subir archivos a GitHub), dime y te guío paso a paso.
- ¡Mucho éxito con el inicio de la Semana 1 mañana! ¿Quieres que detalle otra semana (ej. Semana 5) o necesitas ayuda con algo específico de la Semana 4, como usar Draw.io o escribir consultas SQL?