



Especialización en Bases de Datos

Fundamentos de Bases de Datos Relacionales

Nivel: Básico / Iniciación | Código: UNW-MYS-101 | Duración: 20 horas académicas

uneweb.edu.ve

Programa de Formación en Ingeniería de Datos 2026

MySQL — Nivel I

Fundamentos de Bases de Datos Relacionales

Presentación del curso

Este curso introduce al participante en el mundo de las bases de datos relacionales utilizando MySQL, uno de los sistemas gestores de bases de datos más utilizados a nivel mundial. Desde la instalación del entorno hasta la creación de consultas complejas, el estudiante adquirirá las bases sólidas que le permitirán continuar hacia niveles avanzados de diseño, optimización y administración.

Objetivo general

Capacitar al participante en los fundamentos del modelo relacional y el lenguaje SQL mediante MySQL, para diseñar bases de datos simples, realizar consultas de datos y aplicar operaciones CRUD sobre tablas relacionadas.

Objetivos específicos

- Instalar y configurar el entorno MySQL (MySQL Server, MySQL Workbench).
- Comprender el modelo relacional: tablas, filas, columnas, claves primarias y foráneas.
- Crear y gestionar bases de datos y tablas con sentencias DDL (CREATE, ALTER, DROP).
- Insertar, actualizar y eliminar datos con sentencias DML (INSERT, UPDATE, DELETE).
- Realizar consultas de selección con SELECT, WHERE, ORDER BY, LIMIT y funciones de agregación.
- Combinar datos de múltiples tablas mediante JOINS (INNER, LEFT, RIGHT).
- Aplicar filtros avanzados con GROUP BY, HAVING y subconsultas básicas.
- Importar y exportar datos, y hacer respaldos básicos con mysqldump.

Dirigido a

- Estudiantes y profesionales sin experiencia previa en bases de datos.
- Desarrolladores que necesiten gestionar datos desde sus aplicaciones.
- Analistas de datos que requieran consultar fuentes relacionales.
- Administradores en formación que deseen iniciar en gestión de bases de datos.

Requisitos previos

- Conocimientos básicos de lógica (variables, condicionales).
- Computador con Windows, macOS o Linux, y 4 GB de RAM mínimo.

Metodología

Cada módulo combina exposición teórica con ejercicios prácticos guiados en MySQL Workbench. Se plantean casos de negocio reales (gestión de inventarios, clientes, pedidos) sobre los que se aplican los conceptos de cada sesión. Claude.ai se integra como asistente para generar, revisar y explicar sentencias SQL, acelerando el aprendizaje de la sintaxis y la comprensión del comportamiento del motor.

Plan de contenidos (20 horas académicas)

Módulo	Tema	Horas
1	Introducción a bases de datos relacionales y configuración del entorno MySQL	2
2	Modelo relacional: entidades, atributos, claves y relaciones; Diagrama ER básico	2
3	DDL — Creación y modificación de estructuras: bases de datos, tablas y tipos de datos	3
4	DML — Inserción, actualización y eliminación de registros; integridad referencial	3
5	Consultas SELECT: filtros WHERE, ORDER BY, LIMIT, operadores lógicos y de comparación	3
6	Funciones de agregación: COUNT, SUM, AVG, MIN, MAX; GROUP BY y HAVING	2
7	JOINS: INNER JOIN, LEFT JOIN, RIGHT JOIN y combinación de múltiples tablas	3
8	Subconsultas, vistas, exportación de datos y respaldo con mysqldump; proyecto final	2

Detalle de módulos

Módulo 1 — Introducción y configuración del entorno

- ¿Qué es una base de datos? Diferencias entre bases relacionales, NoSQL y en memoria.
- Historia y posicionamiento de MySQL en el ecosistema de datos.
- Instalación de MySQL Server (Community) y MySQL Workbench en Windows/macOS/Linux.
- Conexión al servidor local; interfaz de MySQL Workbench: esquemas, consola SQL y herramientas.
- Primeros pasos: ejecución de sentencias básicas (SHOW DATABASES, SHOW TABLES).

Módulo 2 — Modelo relacional y diagrama ER

- Conceptos: entidad, atributo, dominio, tupla; tablas como representación de entidades.
- Clave primaria (PK) y clave foránea (FK); restricciones de unicidad y NOT NULL.
- Tipos de relaciones: uno a uno, uno a muchos, muchos a muchos.
- Elaboración de un Diagrama Entidad-Relación (ER) para el caso de negocio del curso.
- Traducción del diagrama ER a esquema de tablas normalizado (1FN y 2FN).

Módulo 3 — DDL: creación y modificación de estructuras

- Sentencias CREATE DATABASE, USE, DROP DATABASE.
- CREATE TABLE: tipos de datos (INT, VARCHAR, TEXT, DATE, DATETIME, DECIMAL, BOOLEAN).
- Definición de PK, FK, UNIQUE, NOT NULL, DEFAULT y AUTO_INCREMENT.
- ALTER TABLE: agregar, modificar y eliminar columnas; cambiar tipos.
- DROP TABLE y TRUNCATE; diferencias y consecuencias sobre los datos.
- Laboratorio: creación del esquema completo para un sistema de gestión de pedidos.

Módulo 4 — DML: manipulación de datos e integridad referencial

- INSERT INTO: inserción individual y múltiple.
- UPDATE con cláusula WHERE; riesgos de actualizaciones sin filtro.
- DELETE con WHERE; diferencias entre DELETE y TRUNCATE.
- Restricciones de integridad referencial: ON DELETE CASCADE, SET NULL, RESTRICT.
- Transacciones básicas: START TRANSACTION, COMMIT y ROLLBACK.

- Laboratorio: carga de datos de prueba y actualización de registros del sistema de pedidos.

Módulo 5 — Consultas SELECT y filtros

- Sintaxis completa de SELECT: columnas, alias (AS), literales.
- Cláusula WHERE con operadores =, !=, >, <, BETWEEN, IN, LIKE, IS NULL.
- Operadores lógicos AND, OR, NOT para combinar condiciones.
- ORDER BY ASC / DESC; LIMIT y OFFSET para paginación.
- Funciones de cadena (CONCAT, UPPER, LOWER, LENGTH) y de fecha (NOW, DATE_FORMAT, DATEDIFF).

Módulo 6 — Funciones de agregación

- COUNT(*) y COUNT(columna): diferencias al tratar NULLs.
- SUM, AVG, MIN y MAX aplicadas a campos numéricos.
- GROUP BY: agrupación de resultados por uno o varios campos.
- HAVING: filtrado de grupos (diferencia con WHERE).
- Combinación de SELECT, GROUP BY y HAVING para reportes de negocio.

Módulo 7 — JOINS y combinación de tablas

- Concepto de JOIN y por qué es necesario en bases relacionales.
- INNER JOIN: intersección de registros que cumplen la condición de unión.
- LEFT JOIN / RIGHT JOIN: conservar registros de la tabla principal aunque no haya coincidencia.
- JOIN de tres o más tablas: encadenamiento de condiciones ON.
- Laboratorio: consultas de reportes cruzando tablas de clientes, pedidos y productos.

Módulo 8 — Subconsultas, vistas, exportación y proyecto final

- Subconsultas en WHERE y en FROM (tablas derivadas).
- Creación y uso de VIEWS para simplificar consultas complejas.
- Exportación de resultados a CSV desde MySQL Workbench.
- Respaldo completo con mysqldump y restauración de base de datos.
- Proyecto final: base de datos de gestión (sistema de pedidos o inventario) con consultas, vistas y documentación.

Evaluación

- Ejercicios prácticos por módulo (40%).
- Quiz de conceptos relacionales y SQL (20%).
- Proyecto final integrador (40%).

Recursos y herramientas

- MySQL Server 8.x (Community Edition) — gratuito.
- MySQL Workbench 8.x como entorno gráfico principal.
- Claude.ai para generación asistida y explicación de sentencias SQL.
- Dataset de práctica: sistema de gestión de pedidos (clientes, productos, pedidos, detalle_pedidos).



Especialización en Bases de Datos

Diseño Avanzado, Optimización y Administración

Nivel: Intermedio / Avanzado | Código: UNW-MYS-201 | Duración: 20 horas académicas

uneweb.edu.ve

Programa de Formación en Ingeniería de Datos 2026

MySQL — Nivel II

Diseño Avanzado, Optimización y Administración

Presentación del curso

Continuación del nivel I, este curso profundiza en el diseño normalizado de bases de datos, la optimización de consultas mediante índices y el plan de ejecución, la programación en el servidor con procedimientos almacenados, funciones, triggers y eventos, así como las tareas fundamentales de administración, seguridad y replicación básica en MySQL.

Objetivo general

Capacitar al participante en el diseño avanzado de esquemas relacionales, la optimización de rendimiento y la programación del lado del servidor en MySQL, habilitándolo para gestionar bases de datos en entornos de producción.

Objetivos específicos

- Aplicar normalización hasta la Tercera Forma Normal (3FN) y reconocer la Forma Normal de Boyce-Codd.
- Diseñar índices eficientes y analizar planes de ejecución con EXPLAIN.
- Crear y gestionar procedimientos almacenados, funciones definidas por el usuario y cursores.
- Implementar triggers para auditoría y validación de reglas de negocio en la base de datos.
- Gestionar usuarios, roles, privilegios y seguridad básica del servidor.
- Configurar y monitorear la replicación maestro-esclavo básica.
- Implementar particionamiento de tablas para grandes volúmenes de datos.
- Utilizar el motor InnoDB en profundidad: transacciones, bloqueos y MVCC.

Requisitos previos

- Haber aprobado MySQL Nivel I (UNW-MYS-101) o demostrar dominio equivalente.
- Familiaridad con sentencias DDL, DML y consultas con JOINS y GROUP BY.

Metodología

Cada módulo incluye un laboratorio guiado sobre una base de datos de producción simulada (sistema bancario o e-commerce). Se analiza el rendimiento antes y después de cada optimización. Claude.ai se utiliza para revisar código SQL complejo, sugerir estrategias de índices y explicar el comportamiento del optimizador de consultas.

Plan de contenidos (20 horas académicas)

Módulo	Tema	Horas
1	Normalización: 1FN, 2FN, 3FN y FNBC; detección y resolución de anomalías	3
2	Índices: B-Tree, Hash, Full-Text; diseño de índices y análisis con EXPLAIN	3
3	Programación en el servidor: procedimientos almacenados, parámetros y control de flujo	3
4	Funciones definidas por el usuario, cursores y manejo de errores con DECLARE HANDLER	2
5	Triggers: BEFORE/AFTER INSERT, UPDATE, DELETE; auditoría y validación de negocio	2
6	Eventos programados (Event Scheduler); vistas materializadas con tablas de resumen	2
7	Administración y seguridad: usuarios, roles, privilegios, SSL y cifrado de datos sensibles	3
8	Replicación maestro-esclavo, particionamiento de tablas y proyecto final de optimización	2

Detalle de módulos

Módulo 1 — Normalización avanzada

- Revisión de 1FN (grupos repetidos) y 2FN (dependencia parcial).
- Tercera Forma Normal (3FN): eliminación de dependencias transitivas.
- Forma Normal de Boyce-Codd (FNBC): casos y diferencias con 3FN.
- Desnormalización controlada: cuándo y por qué sacrificar normalización.
- Laboratorio: auditoría y normalización de un esquema heredado con anomalías.

Módulo 2 — Índices y optimización de consultas

- Estructura interna de índices B-Tree e índices Hash en InnoDB/MyISAM.
- Índice primario (clustered) vs. índices secundarios.
- Índices compuestos: orden de columnas y selectividad.
- EXPLAIN y EXPLAIN ANALYZE: lectura e interpretación del plan de ejecución.
- Estrategias de optimización: covering index, index hints y evitar full-table scans.

Módulo 3 — Procedimientos almacenados

- Sintaxis CREATE PROCEDURE; delimitadores y bloques BEGIN...END.
- Parámetros IN, OUT e INOUT.
- Variables locales, estructuras IF-THEN-ELSE, CASE y ciclos LOOP, WHILE, REPEAT.
- Llamada a procedimientos con CALL y manejo de múltiples resultsets.
- Laboratorio: procedimiento de cierre de mes que calcula totales y actualiza resúmenes.

Módulo 4 — Funciones, cursores y manejo de errores

- CREATE FUNCTION: diferencias con procedimientos; funciones deterministas.
- Cursores: DECLARE, OPEN, FETCH, CLOSE; iteración sobre resultsets.
- DECLARE HANDLER para NOT FOUND y SQLEXCEPTION; propagación de errores.
- SIGNAL y RESIGNAL para lanzar errores personalizados.

Módulo 5 — Triggers

- ¿Qué es un trigger? Casos de uso: auditoría, validación y sincronización.
- BEFORE vs. AFTER; triggers sobre INSERT, UPDATE y DELETE.
- Registros NEW y OLD: acceso a valores antes y después del cambio.
- Limitaciones de triggers en MySQL; orden de ejecución con múltiples triggers.
- Laboratorio: tabla de auditoría que registra quién modificó cada registro y cuándo.

Módulo 6 — Eventos programados y vistas de resumen

- Event Scheduler: activación y configuración del servidor.
- Sintaxis CREATE EVENT: ejecución única y recurrente (EVERY).
- Tablas de resumen como sustituto de vistas materializadas en MySQL.
- Patrones de actualización incremental para reportes de alto rendimiento.

Módulo 7 — Administración y seguridad

- Gestión de usuarios: CREATE USER, ALTER USER, DROP USER.
- Privilegios globales, de base de datos, de tabla y de columna; GRANT y REVOKE.
- Roles en MySQL 8.x: asignación y herencia de privilegios.
- Cifrado de conexiones con SSL/TLS; cifrado de columnas sensibles con AES_ENCRYPT.
- Variables del servidor: max_connections, innodb_buffer_pool_size; monitoreo con SHOW STATUS.

Módulo 8 — Replicación, particionamiento y proyecto final

- Replicación binlog maestro-esclavo: configuración en my.cnf, CHANGE MASTER TO.
- Tipos de particionamiento: RANGE, LIST, HASH, KEY; casos de uso.
- Mantenimiento de particiones: REORGANIZE, ADD, DROP PARTITION.
- Proyecto final: optimización de una base de datos de e-commerce con índices, procedimientos, trigger de auditoría y reporte de eventos.

Evaluación

- Laboratorios prácticos por módulo (40%).
- Quiz de optimización y administración (20%).
- Proyecto final de optimización de base de datos (40%).

Recursos y herramientas

- MySQL Server 8.x (Community), MySQL Workbench 8.x.
- MySQL Shell para scripting administrativo.
- Dataset de producción simulada: e-commerce (>500.000 filas para pruebas de rendimiento).
- Claude.ai para revisión de procedimientos SQL, estrategias de índices y explicación del optimizador.



Especialización en Bases de Datos

Fundamentos y SQL Avanzado con PostgreSQL

Nivel: Básico / Iniciación | Código: UNW-PGS-101 | Duración: 20 horas académicas

uneweb.edu.ve

Programa de Formación en Ingeniería de Datos 2026

PostgreSQL — Nivel I

Fundamentos y SQL Avanzado con PostgreSQL

Presentación del curso

Este curso introduce al participante en PostgreSQL, el sistema gestor de bases de datos objeto-relacional de código abierto más potente y completo disponible hoy. Partiendo desde la instalación, el estudiante avanza hasta el manejo de tipos de datos avanzados, funciones analíticas de ventana y las primeras capacidades NoSQL de PostgreSQL como JSON/JSONB.

Objetivo general

Capacitar al participante en el uso de PostgreSQL como sistema gestor de bases de datos objeto-relacional, dominando su dialecto SQL extendido, sus tipos de datos avanzados y las herramientas de administración básica.

Objetivos específicos

- Instalar y configurar PostgreSQL y pgAdmin 4 en diferentes sistemas operativos.
- Diseñar esquemas utilizando los tipos de datos nativos de PostgreSQL (SERIAL, UUID, ARRAY, JSONB, ENUM, RANGE).
- Construir consultas complejas con CTEs (WITH), funciones de ventana (OVER/PARTITION BY) y DISTINCT ON.
- Manejar datos semiestructurados con los operadores JSON y JSONB.
- Comprender el sistema de esquemas (schemas) y namespaces en PostgreSQL.
- Administrar usuarios, roles y privilegios según el modelo de seguridad de PostgreSQL.
- Realizar respaldos con pg_dump y pg_restore, y monitorear conexiones activas.
- Configurar parámetros básicos de postgresql.conf y pg_hba.conf.

Requisitos previos

- Conocimientos básicos de SQL (nivel equivalente a MySQL Nivel I o similar).
- Nociones de modelo relacional: tablas, claves y relaciones.

Metodología

El curso utiliza un dataset de gestión de una red de bibliotecas públicas como hilo conductor. Cada módulo añade complejidad al esquema y a las consultas, permitiendo al participante ver la evolución natural de una base de datos real. Claude.ai se emplea para explorar diferencias entre PostgreSQL y MySQL, generar consultas CTE y explicar el comportamiento de los tipos JSONB.

Plan de contenidos (20 horas académicas)

Módulo	Tema	Horas
1	Introducción a PostgreSQL: arquitectura, instalación y primeros pasos con psql y pgAdmin	4
2	Tipos de datos nativos: SERIAL, UUID, ARRAY, ENUM, tipos de rango (RANGE) y tipos geométricos básicos	3
3	DDL avanzado: schemas, secuencias, constraints diferidos, dominios y tipos compuestos	3
4	DML y consultas avanzadas: RETURNING, UPSERT (INSERT ON CONFLICT), CTEs (WITH)	3
5	Funciones de ventana: OVER, PARTITION BY, ROW_NUMBER, RANK, LAG, LEAD, SUM OVER	3
6	Soporte JSON/JSONB: operadores, indexación GIN y consultas sobre datos semiestructurados	2
7	Seguridad: roles, usuarios, schemas de seguridad, Row-Level Security (RLS)	2
8	Administración básica: pg_dump/pg_restore, postgresql.conf, pg_hba.conf y proyecto final	2

Detalle de módulos

Módulo 1 — Introducción y configuración del entorno

- Historia de PostgreSQL: origen en POSTGRES (Berkeley) y evolución hasta PostgreSQL 16.
- Arquitectura: proceso postmaster, procesos de backend, shared_buffers, WAL.
- Instalación en Windows, macOS (Homebrew) y Ubuntu; configuración del servicio.
- Interfaz de línea de comandos psql: metacomandos \l, \d, \c, \i y \timing.
- pgAdmin 4: gestión gráfica de esquemas, consultas y monitoreo de conexiones.
- Diferencias clave con MySQL: mayúsculas, comillas dobles vs. simples, NULL handling.

Módulo 2 — Tipos de datos nativos avanzados

- Tipos numéricos: INTEGER, BIGINT, NUMERIC(p,s), REAL, DOUBLE PRECISION.
- SERIAL y BIGSERIAL vs. GENERATED ALWAYS AS IDENTITY (PostgreSQL 10+).
- UUID: generación con gen_random_uuid() y uso como clave primaria.
- ARRAY: declaración, inserción, operadores @>, &&, ANY y unnest().
- ENUM: creación de tipos enumerados y alteración con ADD VALUE.
- Tipos de rango (RANGE): DATERANGE, INT4RANGE y operadores de solapamiento.

Módulo 3 — DDL avanzado y schemas

- Schemas (namespaces): CREATE SCHEMA, search_path y organización multi-tenant.
- Secuencias: CREATE SEQUENCE, NEXTVAL, CURRVAL y SETVAL.
- Constraints diferidos (DEFERRABLE INITIALLY DEFERRED) para cargas masivas.
- Dominios (DOMAIN): validaciones reutilizables con CHECK.
- Tipos compuestos (CREATE TYPE AS) y su uso en funciones y tablas.
- Laboratorio: modelado del esquema de biblioteca con schemas por sucursal.

Módulo 4 — DML avanzado: RETURNING, UPSERT y CTEs

- RETURNING: obtener valores generados tras INSERT, UPDATE o DELETE.
- INSERT ON CONFLICT DO UPDATE (UPSERT) y DO NOTHING; cláusula EXCLUDED.
- WITH (Common Table Expressions): CTEs lectoras y CTEs de escritura (writable CTEs).

- CTEs recursivas: WITH RECURSIVE para jerarquías y grafos de árbol.
- Laboratorio: carga y actualización de catálogo de libros con UPSERT y reporte jerárquico de categorías con CTE recursiva.

Módulo 5 — Funciones de ventana

- Concepto de función de ventana vs. GROUP BY: conservar filas individuales.
- Cláusula OVER y su composición: PARTITION BY y ORDER BY dentro de OVER.
- Funciones de ranking: ROW_NUMBER(), RANK(), DENSE_RANK(), NTILE().
- Funciones de desplazamiento: LAG() y LEAD() para comparaciones temporales.
- Agregados de ventana: SUM, AVG, COUNT con ROWS/RANGE BETWEEN.
- Laboratorio: ranking de préstamos por sucursal y cálculo de tendencia semana a semana.

Módulo 6 — JSON y JSONB

- JSON vs. JSONB: diferencias de almacenamiento, validación y rendimiento.
- Operadores de acceso: -> (object field), ->> (text), #> (path), #>> (text path).
- Funciones: jsonb_each(), jsonb_object_keys(), jsonb_set(), jsonb_strip_nulls().
- Indexación GIN sobre columnas JSONB para búsquedas rápidas.
- Casos de uso: catálogo de metadatos de libros con atributos variables en JSONB.

Módulo 7 — Seguridad y control de acceso

- Modelo de seguridad de PostgreSQL: roles como usuarios y grupos.
- CREATE ROLE, GRANT, REVOKE; herencia de privilegios y LOGIN.
- Privilegios a nivel de schema, tabla y columna.
- Row-Level Security (RLS): CREATE POLICY para acceso por fila según el rol.
- Laboratorio: aislamiento de sucursales mediante RLS (cada rol solo ve sus propios registros).

Módulo 8 — Administración básica y proyecto final

- pg_dump (texto plano y formato custom) y pg_restore para respaldo y recuperación.
- pg_dumpall para respaldo del clúster completo.
- postgresql.conf: parámetros clave (max_connections, work_mem, maintenance_work_mem).
- pg_hba.conf: métodos de autenticación (md5, scram-sha-256, trust, peer).
- Proyecto final: base de datos de biblioteca multi-sucursal con RLS, CTEs de reporte, JSONB para metadatos y respaldo documentado.

Evaluación

- Ejercicios prácticos por módulo (40%).
- Quiz de SQL avanzado y tipos de datos de PostgreSQL (20%).
- Proyecto final de base de datos multi-sucursal (40%).

Recursos y herramientas

- PostgreSQL 16 (Community) — gratuito y de código abierto.
- pgAdmin 4 como entorno gráfico principal.
- DBeaver Community como alternativa multi-base.
- Dataset: red de bibliotecas públicas (libros, autores, sucursales, préstamos, usuarios).
- Claude.ai para explorar diferencias con MySQL, generar CTEs y consultas de ventana.



Especialización en Bases de Datos

Administración, Rendimiento y Extensiones Avanzadas

Nivel: Avanzado / Especialización | Código: UNW-PGS-201 | Duración: 20 horas académicas

uneweb.edu.ve

Programa de Formación en Ingeniería de Datos 2026

PostgreSQL — Nivel II

Administración, Rendimiento y Extensiones Avanzadas

Presentación del curso

Nivel avanzado de la especialización en PostgreSQL. El participante profundiza en la arquitectura interna del motor, la programación del servidor con PL/pgSQL, la extensión del sistema de tipos, la optimización del planificador de consultas, estrategias de alta disponibilidad y el aprovechamiento del ecosistema de extensiones (PostGIS, pg_stat_statements, TimescaleDB).

Objetivo general

Capacitar al participante para administrar, optimizar y extender PostgreSQL en entornos de producción críticos, manejando alta disponibilidad, programación avanzada del servidor y análisis profundo del rendimiento.

Objetivos específicos

- Desarrollar funciones y procedimientos complejos en PL/pgSQL con manejo avanzado de errores.
- Crear triggers de fila y de sentencia para lógica de negocio compleja.
- Diseñar y gestionar tablas particionadas por rango, lista y hash.
- Configurar replicación de streaming (primary-standby) y logical replication.
- Analizar el planificador de consultas con EXPLAIN (ANALYZE, BUFFERS, FORMAT JSON).
- Utilizar índices avanzados: GIN, GiST, BRIN y expresiones parciales.
- Instalar y configurar extensiones clave: pg_stat_statements, PostGIS y TimescaleDB.
- Implementar estrategias de VACUUM, ANALYZE y autovacuum tuning.

Requisitos previos

- Haber aprobado PostgreSQL Nivel I (UNW-PGS-101) o dominio equivalente demostrado.
- Comprensión de schemas, roles, CTEs y funciones de ventana.

Metodología

El curso trabaja sobre una base de datos de telemetría de vehículos conectados (IoT) con millones de registros, ideal para demostrar particionamiento, BRIN, TimescaleDB y optimización del planificador. Claude.ai se emplea para interpretar salidas de EXPLAIN, sugerir estrategias de índices y revisar código PL/pgSQL.

Plan de contenidos (20 horas académicas)

Módulo	Tema	Horas
1	Arquitectura interna: procesos, WAL, MVCC, Visibility Map y el ciclo de VACUUM	3
2	PL/pgSQL avanzado: bloques anónimos, funciones complejas, manejo de errores y cursores	2
3	Triggers avanzados: triggers de fila vs. sentencia, triggers de transición y INSTEAD OF	2
4	Particionamiento declarativo: RANGE, LIST y HASH; sub-particionamiento y mantenimiento	3
5	Índices avanzados: GIN, GiST, BRIN, índices parciales y de expresión; vacuuming y bloat	3
6	Replicación: streaming replication (primary-standby) y logical replication con publicaciones	2
7	Extensiones: pg_stat_statements, PostGIS (geolocalización) y TimescaleDB (series de tiempo)	2
8	Optimización del planificador, tuning de autovacuum y proyecto final de producción	2

Detalle de módulos

Módulo 1 — Arquitectura interna: WAL, MVCC y VACUUM

- Procesos del clúster: postmaster, checkpointer, bgwriter, walwriter, autovacuum launcher.
- Write-Ahead Log (WAL): propósito, segmentos, archivado y punto de control (checkpoint).
- MVCC (Multi-Version Concurrency Control): versiones de fila, xmin/xmax, snapshots.
- Visibility Map y Free Space Map: su rol en VACUUM y el rendimiento de los índices.
- VACUUM vs. VACUUM FULL vs. VACUUM ANALYZE; tablas sin vacuuming: table bloat.
- Configuración de autovacuum: autovacuum_vacuum_scale_factor, cost-based delay.

Módulo 2 — PL/pgSQL avanzado

- Bloques anónimos DO \$\$... \$\$; estructura de funciones PL/pgSQL.
- RETURN, RETURN NEXT y RETURN QUERY para devolver conjuntos de filas (SETOF).
- Manejo avanzado de errores: GET STACKED DIAGNOSTICS, RAISE EXCEPTION con ERRCODE.
- Cursores con y sin parámetros: OPEN, FETCH, CLOSE, MOVE.
- Funciones de ventana definidas por el usuario y funciones de agregado personalizadas (CREATE AGGREGATE).
- Laboratorio: función de cálculo de percentiles de telemetría por vehículo.

Módulo 3 — Triggers avanzados

- Trigger de fila (FOR EACH ROW) vs. trigger de sentencia (FOR EACH STATEMENT).
- Tablas de transición (REFERENCING NEW TABLE / OLD TABLE) en triggers de sentencia.
- Triggers INSTEAD OF sobre vistas actualizables.
- Función de trigger que devuelve NULL para cancelar la operación.
- Laboratorio: sistema de auditoría con tabla de cambios estructurados en JSONB.

Módulo 4 — Particionamiento declarativo

- Particionamiento RANGE: por fecha, por rango numérico; DEFAULT partition.
- Particionamiento LIST: por país, por categoría.
- Particionamiento HASH: distribución uniforme para claves de alta cardinalidad.

- Sub-particionamiento (nested partitioning) para esquemas de dos dimensiones.
- Mantenimiento: ATTACH/DETACH PARTITION, constraint exclusion y partition pruning.
- Laboratorio: tabla de telemetría particionada por mes con detach automático de particiones antiguas.

Módulo 5 — Índices avanzados y table bloat

- GIN (Generalized Inverted Index): uso en ARRAY, JSONB y texto completo (tsvector).
- GiST (Generalized Search Tree): datos geométricos, rangos y búsqueda de similitud.
- BRIN (Block Range Index): ideal para columnas correlacionadas con el orden físico (IoT timestamps).
- Índices parciales (WHERE condition) e índices de expresión (LOWER(email)).
- Monitoreo de bloat con pgstattuple; REINDEX CONCURRENTLY.
- Índices multi-columna: orden de columnas y el impacto en los access patterns.

Módulo 6 — Replicación

- Streaming replication: configuración de primary (wal_level=replica, max_wal_senders) y standby (recovery.conf / standby.signal).
- pg_basebackup para inicializar el standby desde el primario.
- Monitoreo de lag de replicación con pg_stat_replication.
- Logical replication: publicaciones (PUBLICATION) y suscripciones (SUBSCRIPTION) por tabla.
- Casos de uso: lectura en standby, migración de datos entre versiones, CDC (Change Data Capture).

Módulo 7 — Extensiones clave

- pg_stat_statements: análisis de consultas más lentas, más frecuentes y con mayor I/O.
- PostGIS: tipos GEOMETRY y GEOGRAPHY; consultas espaciales (ST_Distance, ST_Within, ST_Intersects); índice GiST espacial.
- TimescaleDB: hypertables, compresión automática, continuous aggregates y políticas de retención.
- pg_cron: programación de tareas dentro de PostgreSQL.
- Laboratorio: consultas de geolocalización de vehículos con PostGIS y agregados de series de tiempo con TimescaleDB.

Módulo 8 — Optimización del planificador y proyecto final

- EXPLAIN (ANALYZE, BUFFERS, FORMAT JSON): lectura del árbol de plan, costos y nodos.
- Estadísticas del planificador: pg_stats, ALTER TABLE ... ALTER COLUMN ... SET STATISTICS.
- Parámetros del planificador: enable_seqscan, enable_nestloop, effective_cache_size, random_page_cost.
- Tuning de postgresql.conf para producción: shared_buffers (25% RAM), effective_cache_size (75% RAM), wal_buffers, parallel workers.
- Proyecto final: optimización de la base de datos de telemetría — particionamiento, índices BRIN+GIN, TimescaleDB, dashboard de rendimiento con pg_stat_statements.

Evaluación

- Laboratorios prácticos por módulo (40%).
- Quiz de arquitectura y administración avanzada (20%).
- Proyecto final de base de datos de telemetría en producción (40%).

Recursos y herramientas

- PostgreSQL 16 con extensiones PostGIS, TimescaleDB, pg_stat_statements y pg_cron.
- pgAdmin 4 y psql para administración y scripting.
- Entorno de pruebas con Docker Compose (primary + standby) para prácticas de replicación.

- Dataset IoT de telemetría vehicular con +5 millones de filas para pruebas de rendimiento.
- Claude.ai para interpretación de EXPLAIN, revisión de PL/pgSQL y estrategias de particionamiento.